T.C.
YALOVA UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

COMPUTER ENGINEERING DEPARTMENT
COMPUTER ENGINEERING PROGRAMME

EFFICIENT ARABIC TEXT CLASSIFICATION USING FEATURE SELECTION TECHNIQUES AND GENETIC ALGORITHM

M.Sc. THESIS

AHMED AL-DULAIMI

ADVISOR: Assist. Prof. Dr. MURAT OKKALIOĞLU

YALOVA
FEBRUBARY 2023

T.C.
YALOVA UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

COMPUTER ENGINEERING DEPARTMENT
COMPUTER ENGINEERING PROGRAMME

EFFICIENT ARABIC TEXT CLASSIFICATION USING FEATURE SELECTION TECHNIQUES AND GENETIC ALGORITHM

M.Sc. THESIS

AHMED AL-DULAIMI
205105028

ADVISOR: Assist. Prof. Dr. MURAT OKKALIOĞLU

YALOVA
FEBRUARY 2023

ETHICAL STATEMENT

In this thesis, which I prepared in accordance with Yalova University Graduate Education Institute Thesis Writing Rules; I have obtained the data, information and documents I have presented in the thesis within the framework of academic and ethical rules, that I have presented all information, documents, evaluations and results in accordance with scientific ethics and morals, that I have cited all the works I have used in the thesis by making appropriate references, that I have not made any changes in the data used, declares that the work I have presented in this thesis is original, I undertake and declare that I accept any legal responsibility that may arise in case of detection of the contrary.

Ahmed Al-Dulaimi

ACKNOWLEDGEMENT

I would like to thank my esteemed advisor, Assist. Prof. Dr. Murat Okkalıoğlu, who helped me throughout my graduate education, always supported me academically and showed me the right path. I am grateful to Assoc. Prof. Dr. Adem Tuncer and Assoc. Prof. Dr. Alper Kürşat Uysal for serving in my dissertation committee.

I would like to thank my family for always being there for me and for always supporting me.

February – 2023 Ahmed Al-Dulaimi

TABLE OF CONTENTS

ETHICAL STATEMENT i
ACKNOWLEDGEMENT iii
TABLE OF CONTENTS v
LIST OF SYMBOLS AND ABBREVIATIONS xi
LIST OF TABLES xv
LIST OF FIGURES xix
ÖZET xxi
ABSTRACT xxv
1. INTRODUCTION 1
1.1. Motivation 1
1.2. Problem Statement And Contribution 2
1.3. Thesis Structure 2
2. LITERATURE REVIEW 5
2.1. Text Mining 5
2.2. Arabic Datasets 5
2.3. Arabic TC 6
3. ARABIC TEXT 9
3.1. Data Growth 9
3.2. Arabic Language 10
3.3 Arabic Grammer 12
3.4. Foreign Words In Arabic Text 14
4. PRELIMINARIES 17
4.1. TC Types 17
4.1.1. Types of labelling 17
4.1.2. Knowledge engineering and ML 18
4.1.3. Hard and ranking TC 18
4.1.4. Document pivoted and category pivoted TC 18
4.1.5. Exogenous and endogenous knowledge 19
4.2. TC System Components 19
4.3. Input Data 20
4.4. Data Pre-Processing 21
4.4.1. Data cleaning 21
4.4.2. Stop words 21

4.4.3. Tokenization 22
4.4.4. Stemming 22
4.5. TW 23
4.5.1. Notation 24
4.5.2. Unsupervised TW schemes 27
4.5.3. Supervised TW schemes 30
4.5.4. Other TW schemes 43
4.6. FS 44
4.6.1. FS vs. construction and extraction 45
4.6.2. Local vs. global FS 45
4.6.3. Filter and wrapper approach 46
4.7. Text Classifiers 47
4.7.1. Statistical approach 47
4.7.2. ML approach 48
4.8. Genetic Algorithm 54
4.8.1. GA for FS 56
4.8.2. GA parameters 57
4.9. Training, Test, And Validation Sets 59
4.10. Evaluation Metrics 61
5. HYBRID FS-GA APPROACH 65
5.1. Input Data 65
5.2. Hybrid FS-GA Approach 67
5.2.1. Filter FS methods 67
5.2.2. GA based wrapper method 68
6. RESULT AND DISCUSSION 73
6.1. Experiment Setup 73
6.1.1. Hardware and software 73
6.1.2. Combination experiments 73
6.2. Dataset Results 74
6.3. Number Of Features 75
6.4. Filter FS Method Results 77
6.4.1. Precision and recall 81
6.4.2. Filtering FS method summary 82
6.5. Hybrid FS-GA Approach Results 83

6.5.1. Precision and recall 86
6.5.2. Hybrid FS-GA approach summary 87
6.6. Hybrid FS-GA Approach With English And Turkish 87
7. CONCLUSION 91
7.1. Discussion 91
7.2. Contributions 92
7.3. Future Work 92
REFERENCES 93
RESUME 103

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOLS

b	: The intercept value
c	: Constant whose value c ≥ 1
M	: Total number of categories in the training data
M_k	: The predictor value
N	: Total number of documents in the training data
S	: Probability of presence of the feature
λ	: Parameter has a value between 5.0 and 9.0 (7.0 as default)

ABBREVIATIONS

ABC	: Artificial bee colony
AI	: Artificial intelligence
BC	: Before christ
BNS	: Bi-normal separation
CC	: Correlation coefficient
CF	: Collection frequency
CF-IDF	: Collection frequency inverse document frequency
CHI2	: Chi-square
CPD	: Categorical proportional difference
DF	: Document frequency
DFS	: Distinguishing feature selection
DOM	: Document object model
DT	: Decision tree
FN	: False negative
FP	: False positive
FS	: Feature selection
GA	: Genetic algorithm
GI	: Gini index
GR	: Gain ratio
GSS	: Galavotti-sebastiani-simi coefficient
GT	: Gini text
GWO	: Gray wolf optimizer
HTML	: Hypertext markup language
ICF	: Inverse category frequency
IDFEC	: Inverse document frequency excluding category
IG	: Information gain
IGM	: Inverse gravity moment
IPA	: International phonetic association
IR	: Information retrieval
LR	: Logistic regression
MI	: Mutual information
ML	: Machines learning
MP	: Member of parliament

MSA	: Modern standard Arabic
NB	: Naive bayes
NGL	: Ng-goh-low coefficient
NLP	: Natural language processing
OR	: Odds ratio
PIDF	: Probabilistic inverse document frequency
PSO	: Particle swarm optimization
RF	: Random forest
RF scheme	: Relevance frequency
SD	: Standard deviation
SVC	: Support vector machine classifier
SVM	: Support vector machines
TC	: Text classification
TF-IDF	: Term frequency- inverse document frequency
TF-PROB	: Probability-based term frequency
TN	: True negative
TP	: True positive
TW	: Term weighting
UAND	: Ultimate Arabic news dataset
URL	: The address of a web page
VSM	: Vector space model
WIDF	: Weighted inverse document frequency

LIST OF TABLES

Table 3.1. The number and rate of Internet users around the world 9
Table 3.2. The five most used languages on the Internet 10
Table 3.3. The way sentences are pronounced in different Arab countries 12
Table 3.4. Different shapes of Arabic letters 12
Table 3.5. The different meanings of the word (عقد) in Arabic 13
Table 4.1. Multi-label TC example 17
Table 4.2. Single-label TC example 18
Table 4.3. An example of Arabic textual data used in TC 20
Table 4.4. Common stop words in Arabic 22
Table 4.5. An example of text tokenization 22
Table 4.6. An example of text stemming 22
Table 4.7. An example of text data after the pre-processing phase 23
Table 4.8. Fundamental classes used in supervised TW schemes 24
Table 4.9. The basic notations 24
Table 4.10. Feature-category confusion matrix example 24
Table 4.11. News example: features frequency in each document 25
Table 4.12. Confusion matrix for the news example (sport category) 25
Table 4.13. Confusion matrix for the news example (politic category) 26
Table 4.14. Confusion matrix for the news example (economy category) 26
Table 4.15. News example: features ranked by CF value 28
Table 4.16. News example: features ranked by DF value 28
Table 4.17. News example: TF, IDF and TF-IDF values 29
Table 4.18. News example: features ranked by CFIDF value 30
Table 4.19. News example: features ranked by global CHI2 value 31
Table 4.20. News example: features ranked by IG value 32
Table 4.21. News example: features ranked by global MI value 33
Table 4.22. News example: features ranked by global OR value 34
Table 4.23. News example: features ranked by GR value 35
Table 4.24. News example: features ranked by IGM value 36
Table 4.25. News example: features ranked by ICF value 36
Table 4.26. News example: features ranked by TF-IDF-ICF value 37
Table 4.27. News example: features ranked by global RF value 37
Table 4.28. News example: features ranked by global IDFEC value 38

Table 4.29. News example: features ranked by GI value 39
Table 4.30. News example: features ranked by GT value 39
Table 4.31. News example: features ranked by DFS value 40
Table 4.32. News example: features ranked by GSS value 41
Table 4.33. News example: features ranked by global NGL value 42
Table 4.33. News example: features ranked by global CPD value 43
Table 4.34. Local and global FS for multilabel classification 46
Table 4.35. Roulette wheel selection method 56
Table 5.1. UAND versions 66
Table 5.2. Population matrix for first generation 68
Table 5.3. Roulette wheel selection method 69
Table 5.4. New population matrix for second generation 71
Table 6.1. The parameter values for GA 74
Table 6.2. F1-score results with UAND versions 74
Table 6.3. F1-score of different number of features found by TW schemes with the linear SVC 76
Table 6.4. Precision, recall, and f1-score values for all TW schemes when used with the four classifiers at different sizes of features 77
Table 6.5. Time required to obtain results using the four classifiers with different number of features 80
Table 6.6. Precision, recall, and f1-score results of hybrid FS-GA approach and filtering method at different sizes of features 83
Table 6.7. F1-score of hybrid FS-GA approach and filter FS with English and Turkish datasets 88

LIST OF FIGURES

Figure 4.1. Components of a TC system 20
Figure 4.2. The difference between the FS and feature extraction 45
Figure 4.3. The difference between (a) filter and (b) wrapper methods 47
Figure 4.4. Various TC techniques 47
Figure 4.5. Supervised TC techniques framework 48
Figure 4.6. Sigmoid function used in LR 49
Figure 4.7. Linear and non-linear SVM 51
Figure 4.8. The structure DT algorithm 52
Figure 4.9. RF algorithm structure 52
Figure 4.10. Chromosome representation 54
Figure 4.11. Data representation type 55
Figure 4.12. GA working principle 55
Figure 4.13. A single-point crossover operator 57
Figure 4.14. Mutation operator (S1 is the old solution and S2 is the new solution.) 57
Figure 4.15. Elite selection strategy from each generation 59
Figure 4.16. Hold out method for training a ML model 60
Figure 4.17. K-fold cross-validation method for training a ML model 60
Figure 5.1. Structure of the web scraping 65
Figure 5.2. Information available on the news site page with html source 66
Figure 5.3. First single-point crossover attempt 69
Figure 5.4. Second single-point crossover attempt 70
Figure 5.5. Second single-point crossover attempt 71
Figure 5.6. Hybrid FS-GA approach 72
Figure 6.1. Performance of TW schemes within a different set of feature sizes 85
Figure 6.2. Average fitness values (f1-score) found in each generation using hybrid FS-GA approach 86
Figure 6.3. Average fitness values (f1-score) found in each generation using hybrid FS-GA approach 89

ÖZNITELIK SEÇME TEKNIKLERI VE GENETIK ALGORITMA KULLANILARAK ETKIN ARAPÇA METIN SINIFLANDIRMASI

ÖZET

Son yıllarda Internet kullanıcılarının sayısındaki önemli artışa, özellikle metin verileri olmak üzere her an paylaşılan veri sayısında çok büyük bir artış eşlik etti. Metin verisi sayısındaki bu büyük artış, araştırmacıların bu büyük miktardaki veriyle başa çıkabilmeleri ve önceden tanımlanmış kategoriler halinde sınıflandırabilmeleri için metin sınıflandırma teknikleri geliştirmelerini gerekli kılmaktadır. Sınıflandırıcıları kullanan metin sınıflandırma teknikleri, terim ağırlıklandırma yöntemleri ve özellik seçme yöntemleri tarafından belirlenen özelliklere dayalı olarak metin belgelerinin hangi sınıflara ait olduğunu belirler. Arapça, İngilizce gibi diğer dillere, Almanca ve İspanyolca gibi Avrupa dillerine ve Çince gibi bazı Asya dillerine kıyasla yeterli miktarda metin sınıflandırma çalışması almayan dillerden biri olarak kabul edilmektedir. Arapça, dünyadaki en çok kullanılan dillerden biri olarak kabul edilir ve İnternet'teki Arapça kullanıcılarının sayısı son yirmi yılda çok büyük bir oranda artmıştır. Arapça ile ilgili araştırma eksikliği birkaç nedenden kaynaklanmaktadır, bunlardan en önemlisi, genel olarak Arapça metinlerle uğraşmanın, Arapça konuşanlar için bile İngilizce gibi diğer yaygın kullanılan dillerde yazılmış metinlerle uğraşmaktan çok daha zor olmasıdır. Arapça, dünyadaki en çok türetilmiş dillerden biridir ve bu, onunla uğraşmayı zor bir görev haline getirir ve onu araştırmacılar için büyük bir meydan okuma haline getirir. Çeşitli metin sınıflandırma teknikleri ve optimizasyon algoritmaları kullanılarak veri toplama, veri işleme ve metin sınıflandırma teknikleri ve yöntemleri zaman içinde geliştikçe, metin sınıflandırma ile ilgili çalışmalar uzun zaman önce başlamıştır. Arapça metinlere gelince, Arapça metinlerin tasnifini ele alan pek çok çalışma olmasına rağmen, çalışmalar tam ayrıntılardan yoksundu ve maksimum fayda sağlamak için birçok tekniği tek bir sisteme entegre etmeye yönelik çok fazla girişim yoktu. Etkili bir metin sınıflandırma modeli oluşturmanın ilk adımı, uygun bir veri kümesi seçmektir. Bu tezde, web kazıma teknikleri kullanılarak yeni bir Arapça veri seti oluşturulacaktır. Yeni veri seti, her versiyona uygulanan farklı ön işleme teknikleri seti ile dört farklı versiyon içeriyor. Dört versiyonun tümü test edilir ve aşağıdaki deneylerde kullanılmak üzere uygun veri versiyonu, en iyi performansa sahip veri versiyonu, seçilir.

Ortak özellik seçim tekniklerinin, filtreleme özellik seçim yöntemi ve sarmal özellik seçim yöntemi, tek bir yaklaşımda kombinasyonuna dayanan geliştirilmiş bir yaklaşım, seçilen veri sürümden en iyi özellikleri seçmek için dört sınıflandırıcı, 21 terim ağırlıklandırma yöntemleri ve genetik algoritma ile iki yöntemin avantajlarından en iyi şekilde yararlanmak için kullanılacaktır. İlk aşamada, filtreleme özellik seçim yöntemi, terim ağırlıklandırma yöntemleri kullanılarak her terime belirli bir değer verilir ve burada en yüksek değere sahip özelliklerin sınıflandırma sürecinde en önemli olduğu varsayılır. En iyi *n* özellik, terim ağırlıklandırma yöntemlerinden elde edilen puanla seçilir ve metin sınıflandırması için sınıflandırıcılarla birlikte kullanılır. Filtreleme özellik seçim yöntemi aşamasının amacı, en iyi sınıflandırıcıyı ve terim ağırlıklandırmanın en iyi 5 yöntemleri seçmektir. Kullanılacak yöntemlerin sayısını azaltmak, özelliklerin en iyi alt kümesini bulmak için genetik algoritma ile kullanmak üzere özenle seçilmiş bir yüksek kaliteli özellikler kümesinin seçilmesine yardımcı olacaktır.

Filtreleme özelliği seçim yönteminin sonuçlar tartışıldıktan sonra, her bir yöntem kullanılarak seçilen özellikler kullanılarak elde edilen sonuçlara göre en iyi terim ağırlıklandırma yöntemleri seçilir. En iyi sınıflandırıcıları seçilirken dikkate alınması gereken en önemli noktalardan biri zaman olduğundan, sonuçlara ve elde etmek için gereken süreye bağlı olarak en iyi sınıflandırıcıları seçilir. En iyi yöntemleri seçildikten sonra GA aşaması başlar.

Sarmal özellik seçiminde GA, birçok kez en iyi terim ağırlıklandırma yöntemleri kullanılarak seçilen en iyi özellikler üzerinde gerçek biyolojik işlemlere benzer, çaprazlama ve mutasyonlar gibi, işlemler gerçekleştirir; burada özellikler, en iyi özellikleri seçmek için her nesilde değiştirilir ve en iyi özellik alt kümesine ulaşmak için en düşük kaliteli özellikleri azaltır. Genetik algoritma, filtreleme özelliği seçim yönteminden farklı olarak, her denemede seçilen öznitelikleri değiştirir, bu da önerdiğimiz modele daha fazla sayıda iyi öznitelik seçme ve belirli öznitelikler tarafından kısıtlanmama fırsatı verir. Boyut olarak orijinal özellik sayısının %1 ila %90'ı arasında değişen en iyi özellik alt kümesini seçeceğiz. Bu geniş özellik yelpazesinin amacı, mümkün olan en yüksek kalitede mümkün olan en az sayıda özelliği seçmek ve filtre özelliği seçim yöntemi kullanılarak bulunan en iyi sonuçları geride bırakmaktır.
Önerilen model, toplam öznitelik sayısının yalnızca %30'una tekabül eden çok daha az sayıda öznitelik kullanarak filtre özelliği seçim yönteminde bulunan en iyi sonuçlardan daha iyi performans gösterebilmiştir. Sonuçlar, önerilen modelin toplam öznitelik sayısının %1-5 kadarını yüksek verimlilikle seçebildiğini göstermiştir. Önerilen sınıflandırma sistemi, toplam özellik sayısının yalnızca %10'unu kullanan sonuçlarının filtreleme özelliği seçim yöntemi kullanılarak elde edilen en iyi sonuçlardan daha iyi olduğu İngilizce gibi diğer dillerle çok etkili olduğunu kanıtlanmış olduğundan, bu iyi sonuçların sadece Arapça ile ilgili olmadığını belirtmekte fayda var.
Okuyucunun bu tezde atılan adımların önemini daha geniş ve kolay bir şekilde anlaması için Arapça hakkında genel bir fikir sahibi olması için, bu tür araştırmalarda alışılmadık olan, Arapça dilini açıklayan tam bir bölüm sunuldu. Bu tez, 20'den fazla terim ağırlıklandırma yöntemlerinin örneklerle desteklenen ayrıntılı bir açıklamasını sunmaktadır. Bu tez, yüksek kalitede yeni bir Arapça veri seti sunar ve tüm versiyonlarında, tüm araştırmacılara ve Arapça metin sınıflandırmasıyla ilgilenenlere sunar. Bu tez, Arapça metin sınıflandırma tekniklerini geliştirmeyi amaçlayan gelecekteki araştırmalara yardımcı olmak için her bir metin sınıflandırmasının ve terim ağırlıklandırma yöntemlerinin performansı, 3 değerlendirme metriği kullanılarak değerlendirildi ve en iyi sonuçları elde etmek için gereken çalışma süresi hakkında ayrıntılı bilgi sağlamaya katkıda bulunur. Bu katkılar, önerdiğimiz yaklaşımın büyük üstünlüğüne ek olarak, bu tezin en önemli katkıları arasındadır.

Anahtar kelimeler: Arapça metin, Genetik algoritma, Öznitelik seçimi, Metin sınıflandırması, Terim ağırlıklandırma yöntemleri

EFFICIENT ARABIC TEXT CLASSIFICATION USING FEATURE SELECTION TECHNIQUES AND GENETIC ALGORITHM

ABSTRACT

The significant growth of the number of Internet users in recent years has been accompanied by a very large increase in the number of data being shared at every moment, specifically text data. This large increase in the number of text data makes it necessary for researchers to try to develop text classification (TC) techniques to be able to deal with this huge amount of data and classify it into pre-defined categories. Arabic is one of the languages that has not received enough TC studies, as dealing with Arabic texts is much more difficult than dealing with texts written in other languages such as English even for Arabic speakers because of its highly derivative nature, which makes it a great challenge for researchers. This thesis aims to establish an effective and integrated Arabic TC system. Web scraping techniques will be used to create a new Arabic dataset that includes many versions. Different pre-processing techniques will be applied to each version before evaluating all versions and selecting the best version to ensure that high quality data is used with the classification system. An improved approach based on the combination of common feature selection (FS) techniques, filter FS method and wrapper FS method, in one approach will be used to make the most of the advantages of the two methods with four classifiers, 21 term weighting (TW) schemes, and genetic algorithm (GA) to select the best features from the selected data version. Filter FS method techniques use the best features found using TW schemes to classify text using classifiers. When all the classifiers and TW schemes have been evaluated, the best classifier and the best 5 TW schemes are selected, which will help to identify a carefully selected set of high-quality features to use with the GA to find the best subset of features. In wrapper FS, the GA performs operations similar to real biological operations, such as crossover and mutations, on the best features selected using best TW schemes many times, where the features are changed in each generation to choose the best features and reduce the lowest quality features to get to the best subset of features. The results show that the proposed approach, hybrid FS-GA approach, can outperform the best results obtained in the filter FS method, a common FS method, using a subset of features with a size of less than 10% of the total feature size and it can also select up to 1% of the original feature size with great efficiency. It is worth noting that these good results are not only related to the Arabic, as the proposed classification system has proven very effective with other languages such as English and Turkish. This thesis presents a new Arabic dataset of high quality and makes it available, in all its versions, to all researchers and those interested in the Arabic TC, and it contributes to provide detailed information about the performance of each TC and TW scheme, and runtime needed to obtain the best results to aid in future research aimed at developing Arabic TC techniques.

Keywords: Arabic text, Feature selection, Genetic algorithm, Text classification, Term weighting

1. INTRODUCTION

The rapid increase in the number of Internet users in recent years, from 2,336 million users in 2012 to 5,382 million users in 2022 [1], has created large amounts of data. Internet users generate 2.5 quintillion[1] bytes of information per day so much that 90% of the information today has been generated in the last two years [2], and amount of data on the Internet has increased nearly 15 times in just 10 years, from 6.5 trillion bytes in 2012 to 97 trillion bytes in 2022, and this rapid growth is likely to continue to reach 175 trillion bytes in 2025 [3]. This data consists of all over the place, such as social networking sites like Twitter which has more than 313 million monthly active users[2] [4], Facebook with more than 2.85 billion monthly active users [5, 6], opinion sites about books, companies or products, news websites, and satellite data. More than 80% of the information that exists today is stored as text data [7], with many of these documents, it is essential to be able to automatically organize documents into categories to facilitate document retrieval and analysis, and this is what automatic text classification (TC) does because it organizes these massive amounts of data by assigns a documents to one or more predefined categories based on its content [8, 9].

1.1. Motivation

Arabic received a much smaller number of TC studies compared to other common languages such as English. Arabic is a Semitic language, and it is considered one of the most widely spoken languages in the world, as it is used by nearly 600 million people around the world as a native or second language [10]. Arabic is considered one of the most grammatically complex languages, and it also includes many forms of writing that are very difficult to dealt with, which has caused a decrease in the number of TC studies about it compared to many other common languages such as English [11] and many European languages and Asian languages [12, 13]. Colloquial Arabic varies greatly even among cities of the same country. Third is the lack of a sufficient number of Arabic texts. In this thesis, we aim to create a detailed reference for Arabic text classification research by creating an Arabic dataset with high efficiency and

[1] The quintillion is one followed by 18 zeros.
[2] Monthly Active Users (MAU) are registered users within the site who log into the site, Twitter, or Facebook for example, at least once in the last 30 days from the measurement date.

classifying it through an effective TC approach based on the combination of several pre-processing techniques, term weighting (TW) schemes, and text classifiers.

1.2. Problem Statement And Contribution

Although much Arabic TC research have been conducted, there are still problems that require trying to find appropriate solutions. Many of the available Arabic datasets suffer from low data quality, as they contain few documents or contain a small number of categories, as most of the available Arabic datasets contain less than 5 categories. In most of the available Arabic datasets, the effect of different data pre-processing methods on performance has not been sufficiently studied due to the lack of tools and libraries dedicated to processing Arabic texts, which makes it difficult to process and clean Arabic texts before using them in classification. Although many feature selection (FS) methods and TC techniques were used in Arabic TC research, there were very few studies that tried to combine these techniques with optimization algorithms to get better performance. The available research suffers from a lack of detailed results, such as not mentioning the time required to obtain results and using a few numbers of evaluation measures.

To solve these problems, we aim to create a huge dataset that includes a data from different sources and various versions where different pre-processing techniques are applied to each version to choose the appropriate version. The new dataset will be made publicly available to researchers. An appropriate data version is used with the proposed model, which consists of several TW schemes, FS methods, classifiers, and genetic algorithm (GA). Detailed information about the performance of each model using various evaluation metrics and the runtime required to get the best results will be provided to aid future research to develop Arabic TC.

1.3. Thesis Structure

The rest of the thesis will be divided into the following chapters,

Chapter 2 – Literature review shows the previous studies conducted on topics related to this thesis such as creating datasets, classifying Arabic texts, and using filter FS methods and wrapper FS methods Arabic texts classification.

Chapter 3 – The Arabic language chapter provides a comprehensive overview of the increase in the number of Arab users on the Internet in recent years, which has caused a significant increase in the number of Arabic documents on the web. This chapter also provides a detailed explanation of the Arabic in terms of history, grammar, number of users and many other details that aim to form an overview of the Arabic.

Chapter 4 – Preliminaries provide an overview of all the techniques that will be used in this thesis. TW and FS techniques are explained in detail, with an example to show how each technique selects features. This chapter also shows common TC techniques, their types, and their important uses in other fields. GA is also explained in terms of how it works, its important parameters, its uses, and many other details.

Chapter 5 – Hybrid FS-GA approach shows a comprehensive view of the stages of our proposed system, and how each of its stages works, starting from creating the dataset and ending with obtaining the results.

Chapter 6 – Result and discussion show the results we obtained from all the methods we used, with a detailed explanation supported by tables and figures for all the results.

Chapter 7 – Conclusion shows some of the concluding observations and contributions made by the thesis and presents some recommendations that may contribute to the development of future TC research.

2. LITERATURE REVIEW

This chapter provides an overview of previous TC studies related to Arabic datasets, and the Arabic TC.

2.1. Text Mining

Data collection process is the first and one of the most important TC steps that directly affect the results, as the data must be of high accuracy, and suitable for the purpose of classification. Researchers and linguists have long discovered the importance of collecting and organizing words into datasets for use in a variety of applications. In the early 1980's, the first large linguistic dataset, the Cobuild corpus, was created that was used to create a new English dictionary [14]. Another popular dictionary is the Oxford Dictionary, which is one of the world's leading dictionaries used in linguistic research [15]. In subsequent years, researchers began to study the semantics of natural languages as these studies aimed to automatically process real datasets [16]. Before starting to study the linguistic semantics of words, the documents were divided into the available categories according to the availability of some words in the document or not. After studying the linguistic semantics, the researchers discovered that the way to deal with words is not to divide them into specific categories, as the word may appear in more than one category but with a different importance score that depends on the number of times the word occurred in a particular category versus the number of times it occurred in the other categories. In recent years, the great development of computers, which help to collect and organize data, and the spread of the Internet, which provided huge amounts of data, helped in the flourishing of studies related to natural language processing (NLP) [17]. Researchers have created many high-quality datasets in many languages. We can find many high-quality datasets in English (For example, in [18], Newsgroup English benchmark is used, in [19], Reuters 21578 English corpus is used) as well as in Asian languages such as Chinese Souhu news corpus [20], and Thai dataset [21].

2.2. Arabic Datasets

In Arabic TC studies, a variety of Arabic datasets were used. In 2001, one of the first Arabic datasets, Arabic newswire corpus, was released, which includes 2,337 text documents collected from several news agencies such as France Presse agency for the

period from 1994 to 2000 [22]. Also in 2001, another Arab dataset, Al-Hayat corpus, was announced, which covers 42,591 articles that were published on the Al-Hayat agency website in 1998 only, divided into 7 different categories [23]. In [24], three Arabic datasets collected from different sources are used, Akhbar-Alkhaleej, Alwatan, and Al-jazeera news dataset. In [25], Al-Hadith corpus is used. Many datasets were also created that deal with news from international news sites such as BBC corpus [26]. Many of the available Arabic datasets suffer from either a lack of data and quality that negatively affects performance, or the need for more pre-processing operations to be fully ready for classification. To get better performance, some researchers have resorted to creating their own datasets instead of using available datasets, but many of these datasets are either private [27, 28], not available to everyone, or given only upon request [29, 30].

2.3. Arabic TC

Although the number of studies on Arabic TC is less than studies on many other languages, there are many important studies that used many TC techniques, TW schemes and pre-processing on Arabic data. In [31], Mesleh presents an Arabic TC approach. The author uses a dataset consisting of nine different categories that have been processed using pre-processing techniques and a TC approach based on support vector machines (SVM) on Arabic text and one TW scheme, which is the chi-square (CHI2). In [32], Mesleh uses a more precise preprocessing process and six TW schemes instead of just one. The results show that the performance of CHI2, ng-goh-low coefficient (NGL), and galavotti-sebastiani-simi coefficient (GSS) is better with SVM. In [33], three classifiers, SVM, Naïve Bayes (NB), and decision tree (DT), are used, but the main contribution is the creation of their dataset and its availability to other researchers. In [34], four classifiers are used, DT, K-nearest neighbors, NB, and multinomial NB, and two TW schemes, CHI2 and IG. In addition to these studies, several Arabic TC studies using many machines learning (ML) techniques is proposed. In [35], the NB method is used to classify an Arabic dataset consisting of 5 categories. Some other classifiers that used in Arabic TC are maximum entropy [36], *k*-nearest neighbors [37], and Rocchio algorithm [38].

All the studies mentioned so far have used a single FS method, filter FS, which is one of the two most popular types of FS as well as wrapper FS. In the filter FS, the best

features are selected, and each feature is given a specific value using the TW schemes. Later, the higher scored features are used in the classification process. In the wrapper FS, a fixed set of features is not selected even if it has the highest values using TW schemes, but a different set of features is selected in each attempt and the best set is chosen based on the performance of the classifier. Many nature-inspired optimization algorithms have been used to find the best solution in the FS process and TC. Many algorithms such as genetic algorithm (GA), gray wolf optimizer (GWO), and artificial bee colony (ABC) are the most used Arabic TC [39-41]. In [42], a classification system based on CHI2, and ABC algorithm has been proposed. In [43], an Arabic FS system based on GWO is proposed.

In this thesis, different methods are used than the methods usually used in Arabic TC research, where a more detailed approach based on the combination of many techniques is built. The proposed approach in this thesis includes the creation of a new Arabic dataset that includes four different versions, where several different pre-processing techniques were applied to each version. All versions are tested, using 3 TW schemes and four classifiers, and the best version is selected to be used in the next part of the proposed approach. 21 TW methods, which may be the largest number of TW schemes ever used in a single Arabic TC research, are used to assign a value to each feature in the selected version, where the features with the highest score are assumed to be the best features. Our proposed approach includes the combination of different FS methods with optimization algorithms, where the filter FS method is used to select the best classifier and five best TW schemes to be used with GA, which selects the best subset of features, which should be the least possible number and the highest possible quality. The proposed approach in this thesis is the first approach that combines this large number of techniques and schemes in one Arabic TC approach.

3. ARABIC TEXT

Arabic used in more than 22 countries as the main language [44, 45], and it is the second language in 58 countries [46]. There are about 168.1 million Arab users of the Internet until 2015 [47] and it is likely that the number of Arab users has reached 237 million users in 2021 [48]. Despite the significant increase in the number of Arab Internet users and, consequently, the increase in the number of Arabic documents that are being circulated, the search for Arabic TC did not progress as is the case in other languages such as English and European languages. This chapter shows the significant increase in the number of Internet users and the number of Arabic users on the Internet. The Arabic will be discussed from several aspects, such as its history and common grammar, with examples given in all cases.

3.1. Data Growth

Huge amounts of data are shared on the web, which has become the main destination for obtaining information, every moment. With this rapid spread, the number of registered domain names has increased every year, reaching 364 million registered domain names at the end of the third quarter of 2021 [49]. This large amount of data requires the continuous development of search, arrangement, and filtering techniques to help users access information more easily and accurately. Table 3.1 shows the number of Internet users around the world, huge numbers that produce data that does not stop at every moment, which makes it very difficult to deal with [50].

Table 3.1. The number and rate of Internet users around the world

World's Continents	Population	Internet Users	Internet Users Rate	Rate of increase in Internet Users between 2000-2022
Asia	4,352,169,960	2,934,186,678	67.4%	2,467%
Africa	1,394,588,547	652,865,628	46.8%	14,362%
Europe	837,472,045	750,045,495	89.6%	614%
South America	664,099,841	543,396,621	81.8%	2,907%
North America	374,226,482	349,582,583	93.4%	223%
Middle East	268,302,801	211,796,760	78.9%	6,378%
Australia	43,602,955	31,191,971	71.5%	309%

Users use search engines to find information more easily and quickly. Search engines collect information, arrange it, index it, and use various techniques to retrieve relevant information quickly when the user requests it. At the beginning of the Internet, most search engines only supported searches written in Latin letters, but with the increase in the number of users speaking other languages, search engines began to support

languages written in non-Latin characters. Between 2000 and 2021, the number of Internet users in the Middle East increased by 6,378%. 70% of the population of Arab countries are using the Internet today, which is in line with the global average of Internet use and is superior to many other regions of the world, such as Africa, where the number of Internet users is 40% of the total population [51]. The number of users using the Arabic on the Internet has grown by more than 9,348% in the last twenty years, which is the largest increase in the number of users of one language in the world in the same period [48], as shown in Table 3.2.

Table 3.2. The five most used languages on the Internet

Five most used languages on Internet	Population	Internet Users	Internet Users Rate	Rate of increase in Internet Users between 2000-2021
English	1,531,179,460	1,186,451,052	77.5 %	742.9 %
Chinese	1,477,137,209	888,453,068	60.1 %	2,650.4 %
Spanish	516,655,099	363,684,593	70.4 %	1,511.0 %
Arabic	447,572,891	237,418,349	53.0 %	9,348.0 %
Portuguese	290,939,425	171,750,818	59.0 %	2,167.0 %

3.2. Arabic Language

Arabic ranks fourth or fifth most widely spoken languages in the world [52]. The inhabitants of many countries, from the Arabian/Persian Gulf in the east to the Atlantic Ocean, speak Arabic as a native or second language. Arabic is also the language in which all Muslims of the world, who are estimated at 1.9 billion people [53, 54], perform their daily prayers.

Although most non-Arabs refer to the Arab term as an umbrella term for a group of relatively similar people of the same ethnic origin, the reality is that an Arab term can be viewed as a cultural term rather than an ethnic term [55]. Arabs look very different, as there are many differences in their appearance and customs, and Arabs speak many dialects that differ even within the same country. Although the Arab word is associated mainly with Muslims, many other religions have been a major part of Arabs, such as Christians and Jews, as Arabic is a major liturgical language for many Christian churches in the Arab world. Many of the most important medieval Jewish religious and intellectual works were also written in Arabic. Today many Christians and Jews live side by side with Muslims, all of whom are called Arabs. Christian Arabs today constitute about 11% of the total population of Syria, and 40% of the population of Lebanon, and large numbers of Jewish Arabs live in Morocco and Tunisia [56, 57]. The status of Arabic rose due to the spread of Islam among countries, as it became the

language of politics, science, and literature for many centuries in the lands ruled by Muslims. Arabic has a direct and indirect influence on many other languages such as Turkish, Persian, and Kurdish.

Arabic is considered one of the greatest modern languages, and it is one of the languages that has deep historical roots. According to researchers, the history of Arabic extends over the centuries, before the advent of the Christian era, specifically to the fourth century BC [58], when it originated in the Arabian Peninsula and spread in the seventh century BC Arabic descended from a dynasty called the Proto-Semitic. The relationship of Arabic with other Semitic languages is very ancient, as Modern Arabic is considered part of the central group of western Semitic languages. The important feature of Arabic is that it retained a many of the original Semitic features, while most Semitic languages could not have the features of their common ancestral language, so many linguists consider Arabic the most Semitic language in terms of the extent to which it retains Proto-Semitic properties [59]. Arabic is a highly organized etymological language in which morphology plays a very important role [60]. Arabic may be one of the most derived languages in the world, as its words are derived from root words and extend with prefixes, and suffixes [61]. Unlike the English language, and languages that use Latin letters in general, in Arabic, the suffixes are not separated from the word but are attached to it, which makes it difficult to use the truncation techniques that are used in other languages. In addition, these letters sometimes match the basic letters of the word, so care must be taken to use different techniques and be of high accuracy to ensure that the original word is obtained without additions.

Arabic consists of three basic forms: classical, MSA, and colloquial. The classic form is the language of the written Qur'an (the book of Islam). MSA is the universal form across Arab countries and is the written language used for official communications and for writing books and newspapers. The MSA is not spoken in any of the Arab countries, as each country uses its own dialect that contains many new words that have been changed from their classical Arabic origin or borrowed from other neighboring dialects. The colloquial form is spoken and written informally among most individuals in their daily communications, and on social networking sites. Unlike classical Arabic, which gives full details of the sentence by choosing the appropriate words, the MSA, and colloquial modern relies on the way words are pronounced, phonemic expressions,

sentence context and diacritics in conveying the meaning of the sentence in the correct form. The method of writing in colloquial Arabic also differs from one country to another, and sometimes the languages differ between regions of the same country. Table 3.3 shows the differences in the way sentences are pronounced in different Arab countries.

Table 3.3. The way sentences are pronounced in different Arab countries

Variety	**Sentence**		
	I love reading a lot	*I didn't find this old book*	*I want to read a book on the history of France*
Literary Arabic (Common spelling)	أحب القراءة كثيرا	لم أجد هذا الكتاب القديم	أريد أن أقرأ كتابا عن تاريخ فرنسا
Literary Arabic (With all vowels)	أُحِبُّ ٱلْقِرَاءَةَ كَثِيرًا	لَمْ أَجِد هٰذَا ٱلْكِتَابَ ٱلْقَدِيمَ	أُرِيدُ أَنْ أَقْرَأَ كِتَابًا عَنْ تَارِيخِ فَرَنْسَا
Classical Arabic (Liturgical or poetic)	uhibu alqira'ata kathira	Lam ajid hatha alkitab alkadhim	Kuntu uredu an akraa kitaban an tarihi faranca
MSA	uhibu alqira'a kathirann	Lam ajid hatha alkitab alkadhim	Kunt ured an akraa kitab an tarih faranca
Iraqi Arabic	Ani ahub lqraiaa kulish	Ma liget hatha elkitab ilkadhim	Gent ared akraa kitab an tarih faranca
Syrian Arabic	Ana ktir bahib lqraee	Ma laqet haeda elkitab elaadem	Kann bedi iraa kitab an tarih franca
Egyptian Arabic	Ana baheb lqeraia awi	Maletesh elktab elaadem da	Ana kunt awez aara ktab an tarih franca

3.3 Arabic Grammer

The alphabets of Arabic consist of 28 letters (ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ظ ط ع غ ف ق ك ل م ن ه و ي) in addition to the hamza (ء). There are no uppercase and lowercase letters in Arabic as in other languages, as it is written from right to left. Table 3.4 shows the letters of Arabic alphabet with their way of writing them in their four states: separate (S), as a first (F), middle (M), and last (L) letter in the word, along with the international phonetic association (IPA) symbols for Arabic letters.

Table 3.4. Different shapes of Arabic letters

S	**F**	**M**	**L**	**IPA**	**S**	**F**	**M**	**L**	**IPA**	**S**	**F**	**M**	**L**	**IPA**
ا	ا	ـا	ـا	/aa/	س	سـ	ـسـ	ـس	/s/	ل	لـ	ـلـ	ـل	/ɬ/
ب	بـ	ـبـ	ـب	/b/	ش	شـ	ـشـ	ـش	/ʃ/	م	مـ	ـمـ	ـم	/m/
ت	تـ	ـتـ	ـت	/t/	ص	صـ	ـصـ	ـص	/sˤ/	ن	نـ	ـنـ	ـن	/n/
ث	ثـ	ـثـ	ـث	/ɵ/	ض	ضـ	ـضـ	ـض	/dˤ/	ه	هـ	ـهـ	ـه	/h/
ج	جـ	ـجـ	ـج	/ʒ/	ط	طـ	ـطـ	ـط	/tˤ/	و	و	ـو	ـو	/w/
ح	حـ	ـحـ	ـح	/ɦ/	ظ	ظـ	ـظـ	ـظ	/ðˤ/	ي	يـ	ـيـ	ـي	/j/
خ	خـ	ـخـ	ـخ	/x/	ع	عـ	ـعـ	ـع	/ʕ/	ء	-	-	-	/ʔ/
د	د	ـد	ـد	/d/	غ	غـ	ـغـ	ـغ	/ɣ/	ة	-	-	ـة	/t/
ذ	ذ	ـذ	ـذ	/ð/	ف	فـ	ـفـ	ـف	/f/	ى	-	-	ـى	/aa/
ر	ر	ـر	ـر	/r/	ق	قـ	ـقـ	ـق	/q/	-	-	-	-	-
ز	ز	ـز	ـز	/z/	ك	كـ	ـكـ	ـك	/k/	-	-	-	-	-

What distinguishes Arabic, and increases its difficulty and complexity, is the presence of many diacritics that change the way the word is pronounced lightly, causing the entire meaning of the word to be changed. Arabic text does not provide enough information about the correct pronunciation of the word, so many diacritics and lengthening rules are used, which provides an audio guide that gives enough information to the person about the way the word is pronounced and the correct way to use it. The vowels in Arabic affect the meaning of the word completely, as writing words without using the vowels, and without using the word in a specific sentence, may lead to reading it completely differently. Table 3.5 shows the different meanings of one word (عقد).

Table 3.5. The different meanings of the word (عقد) in Arabic

Meaning in English	Meaning in English	Example in Arabic	The form of the word in Arabic
necklace	gold necklace	عِقدٌ ذهبي	عِقدٌ
Holding a conference	The president held a conference	عَقدَ الرئيس مؤتمراً	عَقدَ
Decade	A decade has passed	مر عَقدٌ من الزمن	عَقدٌ
knot	The rope is knotted tightly	عُقِدَ الحبل بقوة	عُقِدَ

Tatweel is a Arabic typesetting which cannot be considered an Arabic letter [23]. This addition can be inserted between any two consecutive letters, for example "هرب" (*/harrab/ <run away>*) which can be written as "هـرب" or "هـرب" or even "هــرب". In all the previous forms, the meaning of the word does not change.

Arabic words are generally classified into three main categories: nouns, verbs, and particles. A noun is a word that has a meaning without any association to time and is divided into definite or indefinite nouns. One of the most direct affixes to a defining noun is the defining term "ال" (*/al/<the>*), personal pronouns such as "انا" (*/ana/<I>*), "انت" (*/ant/<You>*), "نحن" (*/nahn/<We>*), and demonstrative pronouns such as "هذا" (*/haðhaa/<this-masculine>*), "هذه" (*/haðihi/<this-feminine>*).

A verb is a word that refers to an action that occurred by a person or thing at a specific time. One example of a verb in Arabic is the sentence "كَتَب الطالبُ الواجبَ في الصَباحِ" (*/kataba altˁalbu aloajeba fi alsabahi/<the student wrote the homework in the morning>*), where "كَتَب" (*/kataba/<wrote>*) is the verb that refers to a specific person, which is the "الطالبُ" (*/altˁalbu/<student>*) who actually did the "الواجبَ" (*/aloajeba/<the homework>*) at a specific time, "في الصَباحِ" (*/fi alsabahi/<In the morning>*).

Words that are neither nouns nor verbs are considered particles. Particles are not words in themselves, but the meaning of the sentence is not complete without their presence, for example prepositions, such as "من" (*/min/<from>*), "الى" (*/ela/<to>*), "في" (*/fi/<in>*), and conjunctions, such as "ثم" (*/θma/<then>*), "او" (*/aau/<or>*).

Arabic has two genders, usually referred to as masculine and feminine. A word ending in "ة" (*/t/*) usually refers to the feminine. For example, "مهندس" (*/muhandis/<engineer-masculine>*) is a singular masculine word, if we add the letter "ة" at the end of the word it becomes "مهندسة" (*/muhandisa/<engineer-feminine>*), which is the feminine singular form of the word.

3.4. Foreign Words In Arabic Text

The world's languages have exchanged many words with each other since ancient times. Previously, words were mainly exchanged through trade because of the movement of merchants to distant countries and the need to learn some words to talk to the natives who traded with them. Recently, with the development of the Internet and technology, the exchange of words has become easier. When exchanging words between languages, new meanings are usually given new words that correspond to the conventions of the target language. There are some words such as scientific names, technical terms and names of new technological inventions that are not easily translated and if the translation is done, the new word does not express the original word clearly, so it is preferable to use the original word unchanged with changing the way the original word is pronounced to phonemic sounds to the target language through transliteration. Not all the phonemes of the source language may be present in the target language, so some rounding is often necessary, and this problem occurs not only in languages that use different letters but even in languages that use similar letters, an example of this is the difference in writing "Color" and "Colour" across American and British usage.

There are some words that have their origin in other languages, such as the words "فيروس" (virus), "ارشيف" (archive) and "راديو" (radio), but they have been used frequently and for a long time, so they have one version in Arabic and a semi-unified way of writing.

In the age of technology and the Internet, there are large numbers of non-Arabic words that enter Arabic and are used by the original users daily before a standardized form for these words is determined. English sounds not found in Arabic include */p/*, */r/*, */ʒ/*, */g/*, and */n/* resulting in multiple phonemic translations of the same word in English. For example, if Gorbachev, the head of state in the former Soviet Union, is mentioned in one of the news, his name will make a big difference between different newspapers as his name is not Arabic and the first letter of it is */g/* which is one of the English sounds not found in Arabic, so It is usually replaced by one of Arabic letters such as "غ" /ɣ/, "ق" /q/, "ج" /ʒ/ and "ك" /k/ so that the name "Gorbachev" is in one of the following forms: "غورباتشوف", "جورباتشوف", "قورباتشوف" and "كورباتشوف" [62].

4. PRELIMINARIES

TC is the process of determining the appropriate category to which text documents belong based on its content. In other words, it is similar to reading documents, understand the target it refers to, and then selecting the appropriate category [63]. This chapter gives an overview of all the techniques that will be used in this thesis, the concept of TC, data collection and processing method, TW schemes and the classifiers and algorithms that we will use for TC and FS.

4.1. TC Types

4.1.1. Types of labelling

There are many applications where TC are used, some of these applications have only two categories to choose from and other application can contain (0-*k*) categories. In multi-label TC (also called overlapping categories), some documents can belong to more than one category at the same time, while in single label TC the classifier categorizes each document into exactly one category, although there are usually several categories from which to choose. In Binary TC, only two categories are available, and all documents should be classified into one of these two categories only [64]. This type is very important because it is used as a sub-action in multi-label TC and single-label TC [65], and used in many applications, such as determining negative and positive reviews. Table 4.1 shows an example of multi-label TC and Table 4.2 shows an example of single label TC.

Table 4.1. Multi-label TC example

Title	Label
أعلنت محكمة كاتانيا افلاس نادي كاتانيا الإيطالي وهبوطه الى دوري الدرجة الثالثة الايطالي وتعيين وصي على النادي لحين اكمال اجرائات بيعه في المزاد. ***Translate as,*** The Catania Court announced the bankruptcy of the Italian club, Catania, and its relegation to the Italian third division, and the appointment of a trustee for the club until the procedures for selling it at auction are completed.	Sport, Economy
على الرغم من الغزو الروسي، أعلنت شركة "انتل" انها لا تتوقع انخفاضاً كبيراً في أسهم الشركة حيث ان لديها حلولاً بديلة لروسيا وأوكرانيا فيما يخص المواد الخام لصناعة منتجات الشركة. ***Translate as,*** Despite the Russian invasion, Intel Corporation announced that it does not expect a significant decline in the company's shares, as it has alternative solutions to Russia and Ukraine in terms of raw materials to manufacture the company's products.	Politic, Economy, Technology

Table 4.2. Single-label TC example

Title	Label
أعلنت محكمة كاتانيا إفلاس نادي كاتانيا الإيطالي القابع في الدرجة الثالثة بسبب الإفلاس، وقد أعلنت المحكمة عن تعيين وصي على النادي لحين انتهاء المزاد. ***Translate as,*** The Catania court declared the Italian club, Catania, in the third degree, bankrupt due to bankruptcy, and the court announced the appointment of a trustee for the club until the end of the auction.	Sport
أعلنت شركة فايزر أنها أطلقت اختبارات سريرية في مراحل متوسطة إلى متقدّمة، لحبّة تقي من كوفيد-19. ***Translate as,*** Pfizer announced that it has launched clinical tests, in intermediate to advanced stages, for a pill that protects against Covid-19.	Medical

4.1.2. Knowledge engineering and ML

Until the late 1980s, TC was done in a knowledge engineering method. Expert systems are created by experts and engineers, where fixed rules and specific steps are established that the system uses to classify documents with sufficient accuracy. At the beginning of the 1990's, with the development of computing power, the ML approach gained great popularity. In the ML, systems rely on artificial intelligence (AI) to automatically determine which categories documents belong to instead of using specific rules. The classifier is trained using several predefined data and categories, which are called training data, before the system is tested using test data that is without a predefined category. ML techniques, in addition to their high speed, are characterized by their great ability to adapt, and quickly retrain when categories change.

4.1.3. Hard and ranking TC

After entering the use of ML, knowledge engineers and experts began to use ML as an auxiliary technology when creating knowledge systems grammar-based, or anyone who categorizes texts manually, they use ML to find out the number of documents, the number of categories, the number of documents that belong to each category, and so on. This is called a text ranking arrangement [63]. On the other hand, others have started to use a system based entirely on ML techniques. These systems are called hard TC systems. In this thesis hard TC systems will be used.

4.1.4. Document pivoted and category pivoted TC

Since most of the text data that is dealt with consists of two main parts: the document, which shows the text contents, and the category, which indicates the category to which the document belongs. TC can be implemented in two different ways. The first method

is the pivot document TC, where all classes to which the document belongs are found [66]. The second method is the category pivoted TC, where all the documents that belong to a particular category are found [63]. The data and the categories are treated in the same way regardless of the method used, as the traditional form of the data should not change with the change of classification technology. Although the two methods may look very similar, they are often used in different applications as each offers better performance. Document pivoted TC is appropriate in stable systems to which data is added sequentially and regularly, such as systems that classify incoming messages to e-mail as important or not, while category pivoted TC is appropriate when we have a ready set of data to categorize them, or when we add new categories to an existing system [67].

4.1.5. Exogenous and endogenous knowledge

Some systems may use external knowledge sources to help in TC, they use additional data, in addition to the content of the document itself, to help classify the document into a specific category. Most of the data that is used contains a lot of information about the document, such as author's name, the publisher, the date of publication, and keywords. In most cases, the most important data is ignored and only the content and category used, but if we want to outsource, we can use other information that was ignored to help make the appropriate decision for classification. This type called exogenous knowledge text classification. The other type is endogenous knowledge text classification, which uses the content of the document itself, as the document content often contains many terms that refer to self-knowledge, so it can be used to classify accurately without the need for an external source [63, 68, 69]. In this thesis, FS methods and TC methods that use endogenous knowledge only are utilized.

4.2. TC System Components

The TC system consists of many components that complement one another, so care must be taken that every step is done as best as possible by using the best methods, starting with the data that must be a high quality data, the stage of pre-processing, in which care must be taken that the data is in the best possible form, and that it is free of any unnecessary structures or terms that would decrease the model performance, up to

the stage of representing text documents to be ready to be dealt with through classifiers. Figure 4.1 shows the components of a TC system.

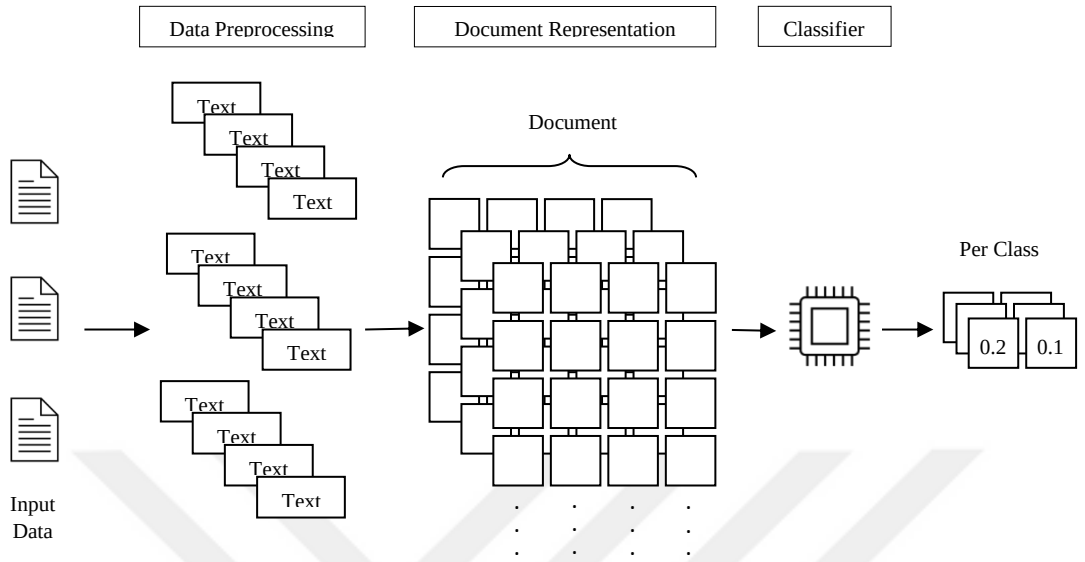

Figure 4.1. Components of a TC system

4.3. Input Data

Since the proposed approach in this thesis aims to classify the text, the data that will be entered is textual data. Table 4.3 shows an example of the data that is used in single-label TC, where each document consists of the text and the category it refers to, as each document belongs to only one category. Since this thesis is about the Arabic, the Arabic was used in the example shown in Table 4.3, but in general any other language can be used in TC.

Table 4.3. An example of Arabic textual data used in TC

Sentence before data Pre-Processing	**Category**
أعلنت رئيسة الوزراء البريطانية، ليز تراس، استقالتها بعد 45 يوماً فقط من استلامها للمهمة، لتنهي بذلك أقصر فترة حكم في تاريخ بريطانيا. *Translated as:* British Prime Minister Liz Terrace announced her resignation just 45 days after taking office, ending the shortest term in British history.	Politic
عزز نادي ارسنال من صدارته للدوري الانكليزي الممتاز بعد انتصاره على توتنهام في الجولة العشرون من البريمرليغ. سجل ارسنال هدفي الفوز في الشوط الاول في الدقيقتين 14 و36. *Translated as:* Arsenal strengthened its lead in the English Premier League, after its victory over Tottenham in the 20th round of the Premier League. Arsenal scored the winning goals in the first half in the 14th and 36th minutes.	Sport
سجلت اسعار النفط ارتفاعاً ملحوظاً حيث ارتفعت العقود الآجلة للخام الامريكي بنسبة 0.27% لتصل الى 71.31 دولار للبرميل الواحد، للمرة الأولى منذ اكثر من سنتين. *Translated as:* Oil prices recorded a noticeable rise, as US crude futures rose 0.27% to $71.31 per barrel, for the first time in more than two years.	Economy

4.4. Data Pre-Processing

Since the thesis is about Arabic, this section describes the pre-processing methods that are uses with Arabic to ensure that it is free of terms, symbols, and words that do not add meaning to the text, which will help in obtaining more accurate results. The processing methods described in this section can be generalized to any language in proper manner.

4.4.1. Data cleaning

Data cleaning is the first step in dealing with text data which include removing all words, letters and numbers from other languages and removing links, duplicate names, and symbols. The data cleaning process includes several steps, including,

1. An equal number of documents is selected for each category, so that we can get the results as accurately as possible and so that the amount of data for a particular category is not more than the rest of the categories.
2. Get rid of forms that do not contain information or unclassified forms that may cause some problems in TC process.
3. Remove all non-Arabic letters, numbers, punctuation, links, and special marks.
4. Removing the elongation marks for Arabic words that do not add any meaning to the words except by repeating the letters and giving them larger than their real size, such as the word (جـمــيل/beautiful) which becomes (جميل) and the word (قووي/Strong) which becomes (قوي) with the same meaning remaining.
5. Removing the different forms of the same letter, where the letters (أ،إ،آ) are converted to (ا), and (لأ، لإ، لآ) is converted to (لا).

4.4.2. Stop words

The texts include many words that do not contain any significance worthy of being used in the classifiers. Because of the many words in the lexicon and the use of pronouns separately in Arabic, Arabic has many stop words that must be removed from texts and documents [70]. Table 4.4 shows some common stop words in Arabic that are removed from the texts.

Table 4.4. Common stop words in Arabic

Stop Words	Meaning in English
هؤلاء	These
آنفا	Above
اعطى	Gave

4.4.3. Tokenization

This method breaks down sentences into words, symbols, or other meaningful elements called tokens [71]. Table 4.5 shows an example of this process.

Table 4.5. An example of text tokenization

Sentence before tokenization	Sentence after tokenization
قرر الاب اخذ عائلته الى حديقة الحيوانات بعد الانتهاء من انجاز عمله Translated as: The father decided to take his family to the zoo after completing his work	{"قرر"، "الاب"، "اخذ"، "عائلته"، "الى"، "حديقة"، "الحيوانات"، "بعد"، "الانتهاء"، "من"، "انجاز"، "عمله"}

4.4.4. Stemming

Stemming is the process of removing suffixes or prefixes from words to obtain the remaining stem or root of the word, which is not necessarily the origin of the word [72]. The importance of this process that it makes dealing with texts less dependent on the shape of the words, and it also reduces the potential size of vocabulary, which gives us fewer but more accurate features [73]. Table 4.6 shows an example of text stemming technique, and Table 4.6 shows what the documents in Table 4.7 look like after all the pretreatment techniques mentioned in this section have been applied to them. One note, from Table 4.7, the effect of applying all pre-processing techniques on the text, where the data was cleaned as all suffixes, numbers, symbols and stop words were removed and word roots were obtained to get rid of different forms of words and rely on the root. The benefit of these techniques is to reduce the number of features that are dealt with and to keep the largest number of words that are supposed to be more important in the classification process.

Table 4.6. An example of text stemming

The word	Meaning in English	Stemming of the word	Meaning of stemming word
الساعة	The clock	ساعة	Clock
والمعلمون	And the teachers	معلم	Teacher
لاعبينا	Our players	لاعب	Player

Table 4.7. An example of text data after the pre-processing phase

Sentence after data Pre-Processing	Category
أعلن رئيس وزراء بريطانية ليز تراس استقال يوم استلام مهمة تنهي أقصر فترة حكم تاريخ بريطانيا *Translated as:* British Prime Minister Liz Terrace announce resign day take office end short term British history	Politic
عزز نادي ارسنال صدارة دوري الانكليزي ممتاز انتصار توتنهام جولة بريمرليغ سجل هدف فوز شوط الاول دقيقة *Translated as:* Arsenal strengthen lead English Premier League after victory Tottenham round score win goal first half minute	Sport
سجلت اسعار نفط ارتفاع ملحوظ ارتفعت عقود آجلة خام امريكي بنسبة دولار برميل واحد *Translated as:* oil price record notice rise crude future rose barrel	Economy

4.5. TW

We can define a feature as some specific characteristic that is selected from the data that is used. For text documents, the words, terms, and symbols that appear in the documents can be considered as features that can be selected based on their importance. Getting good results depends on choosing good features. Good features can be considered as the features that a classifier can use to determine the category to which the documents belong. Words that appear in one category more often than others are of greater importance to that category. Bad features are those that do not provide good information about documents. It is important to try to remove stop words, terms and phrases that do not add much importance to classification in the data preprocessing step, which will lead to more accurate results.

ML algorithms cannot deal with text data directly, so the data is processed and converted into a comprehensible form for ML algorithms by representing it in a vector space model (VSM), where the contents of the document are represented as a vector $W^j = \{w_1^j, w_2^j, \dots, w_n^j\}$ where w_i^j denotes the weight of each term in the document, this weight indicating the importance of the term in the document. The weights of each term-document couple are assigned according to a predetermined TW scheme, which should purposefully estimate the importance of each term in each document. Term weighting schemes take many factors into account while determining the value of terms and documents, where all documents are treated with the same importance as some short documents may be more important than long documents, which may contain a lot of words that do not provide much importance and most of them are removed in the pre-processing stage. The repetition of a specific term in a document is not related to the content of the category in general but indicates the importance of the term in the document itself. Uncommon terms are usually the most important in a

document, and best express the document's content [2]. Many TW schemes have been studied, where an attempt is made to find the best features while maintaining the small dimension of the data. Table 4.8 shows the confusion matrix, defines the four basic information components of the TW schemes, and Table 4.9 shows the basic notations which will allow us to better understand the TW schemes [74].

Table 4.8. Fundamental classes used in supervised TW schemes

Data class	Class classified as		
	C	$\bar{C}$	
F	$True\ Positive\ (TP)$	$False\ Positive\ (FP)$	N_F
$\bar{F}$	$False\ Negative\ (FN)$	$True\ Negative(TN)$	$N_{\bar{F}}$
	N_C	$N_{\bar{C}}$	

Table 4.9. The basic notations

Notations	Value	Meaning
TP	$Count(F, C)$	The number of documents belonging to class *C* provided the feature F occurs at least once.
FP	$Count(F, \bar{C})$	The number of documents belonging to class *C* and in which the feature F has not occurred.
FN	$Count(\bar{F}, C)$	The number of documents that do not belong to class *C* provided the feature F occurs at least once.
TN	$Count(\bar{F}, \bar{C})$	The number of documents that do not belong to class *C* and in which the feature F has not occurred.
N	*TP+FP+FN+TN*	Number of documents in corpus.
N_F	*TP+FP*	The sum of the number of TPs and FPs.
$N_{\bar{F}}$	*FN+TN*	The sum of the number of FNs and TNs.
N_C	*TP+FN*	The sum of the number of TPs and FNs.
$N_{\bar{C}}$	*FP+TN*	The sum of the number of FPs and TNs.

4.5.1. Notation

When considering different methods of TW, the confusion matrix, shown in Table 4.6, is one of the most important steps to use. Table 4.10 shows the feature-category confusion matrix example, where the values refer to the number of documents.

Table 4.10. Feature-category confusion matrix example

	$v_{Category} = 1$	$v_{Category} = 0$	
$v_{Feature} = 1$	$TP = 290$	$FP = 133$	$N_F = 423$
$v_{Feature} = 0$	$FN = 11$	$TN = 201$	$N_{\bar{F}} = 212$
	$N_C = 301$	$N_{\bar{C}} = 334$	$N = 635$

Where $v_{Feature} = 1$ means that the feature is present in the document, $v_{Feature} = 0$ indicates that the term is not present in the document. $v_{Category} = 1$ means the document belongs to the category, and $v_{Category} = 0$ means the document does not belong to the category. These four values of $v_{Feature}$ and $v_{Category}$ make up the four counts of TP, FP, FN, and TN. The values of $v_{Feature}$ and $v_{Category}$ were summed in each line and column, where we notice that the summation results are shown on the right and bottom of the table, respectively, which gives us five other values, namely

N_F, $N_{\bar{F}}$, N_C, $N_{\bar{C}}$, and the total N. In this section, example consisting of nine documents divided into three categories will be used to understand how TW schemes work, as shown in Table 4.11. The documents in this example are very simplified and are completely different from the real applications documents. Five features will be used which is *ball*, *parliament*, *point*, *member of parliament* (MP) and *barrel*. Binary weights will be used for clarity, and the weights in Table 4.11 will be used to represent weights as in Tables 4.12, 4.13, and 4.14. The documents will be divided into three categories, four documents belong to the sports category, three documents belong to the politics category, and two documents belong to the economy category.

Table 4.11. News example: features frequency in each document

Document	**Number of occurrences for the available features**					**Category**
	'Ball'	*'Parliament'*	*'Point'*	*'MP'*	*'Barrel'*	
#1	1	0	1	0	0	Sport
#2	2	1	1	0	0	Sport
#3	3	0	0	0	0	Sport
#4	1	0	3	0	0	Sport
#5	0	1	0	1	1	Politic
#6	0	1	0	1	0	Politic
#7	0	0	0	3	1	Politic
#8	1	1	0	0	1	Economy
#9	0	0	0	2	1	Economy

Table 4.12. Confusion matrix for the news example (sport category)

	$v_{Category} = v_{\text{Sport}} = 1$	$v_{Category} = v_{\text{Sport}} = 0$	
$v_{Feature} = v_{Ball} = 1$	$TP = 4$	$FP = 1$	$N_F = 5$
$v_{Feature} = v_{Ball} = 0$	$FN = 0$	$TN = 4$	$N_{\bar{F}} = 4$
	$N_C = 4$	$N_{\bar{C}} = 5$	$N = 9$

(a) 'Ball' feature in 'Sport' category

	$v_{Category} = v_{\text{Sport}} = 1$	$v_{Category} = v_{\text{Sport}} = 0$	
$v_{Feature} = v_{Parliament} = 1$	$TP = 1$	$FP = 3$	$N_F = 4$
$v_{Feature} = v_{Parliament} = 0$	$FN = 3$	$TN = 2$	$N_{\bar{F}} = 5$
	$N_C = 4$	$N_{\bar{C}} = 5$	$N = 9$

(b) 'Parliament' feature in 'Sport' category

	$v_{Category} = v_{\text{Sport}} = 1$	$v_{Category} = v_{\text{Sport}} = 0$	
$v_{Feature} = v_{Point} = 1$	$TP = 3$	$FP = 0$	$N_F = 3$
$v_{Feature} = v_{Point} = 0$	$FN = 1$	$TN = 5$	$N_{\bar{F}} = 6$
	$N_C = 4$	$N_{\bar{C}} = 5$	$N = 9$

(c) 'Point' feature in 'Sport' category

	$v_{Category} = v_{\text{Sport}} = 1$	$v_{Category} = v_{\text{Sport}} = 0$	
$v_{Feature} = v_{MP} = 1$	$TP = 0$	$FP = 4$	$N_F = 4$
$v_{Feature} = v_{MP} = 0$	$FN = 4$	$TN = 1$	$N_{\bar{F}} = 5$
	$N_C = 4$	$N_{\bar{C}} = 5$	$N = 9$

(d) 'MP' feature in 'Sport' category

	$v_{Category} = v_{\text{Sport}} = 1$	$v_{Category} = v_{\text{Sport}} = 0$	
$v_{Feature} = v_{Barrel} = 1$	$TP = 0$	$FP = 4$	$N_F = 4$
$v_{Feature} = v_{Barrel} = 0$	$FN = 4$	$TN = 1$	$N_{\bar{F}} = 5$
	$N_C = 4$	$N_{\bar{C}} = 5$	$N = 9$

(e) 'Barrel' feature in 'Sport' category

Table 4.13. Confusion matrix for the news example (politic category)

	$v_{Category} = v_{Politic} = 1$	$v_{Category} = v_{Politic} = 0$	
$v_{Feature} = v_{Ball} = 1$	$TP = 0$	$FP = 5$	$N_F = 5$
$v_{Feature} = v_{Ball} = 0$	$FN = 3$	$TN = 1$	$N_{\bar{F}} = 4$
	$N_C = 3$	$N_{\bar{C}} = 6$	$N = 9$

(a) 'Ball' feature in 'Politic' category

	$v_{Category} = v_{Politic} = 1$	$v_{Category} = v_{Politic} = 0$	
$v_{Feature} = v_{Parliament} = 1$	$TP = 2$	$FP = 2$	$N_F = 4$
$v_{Feature} = v_{Parliament} = 0$	$FN = 1$	$TN = 4$	$N_{\bar{F}} = 5$
	$N_C = 3$	$N_{\bar{C}} = 6$	$N = 9$

(b) 'Parliament' feature in 'Politic' category

	$v_{Category} = v_{Politic} = 1$	$v_{Category} = v_{Politic} = 0$	
$v_{Feature} = v_{Point} = 1$	$TP = 0$	$FP = 3$	$N_F = 3$
$v_{Feature} = v_{Point} = 0$	$FN = 3$	$TN = 3$	$N_{\bar{F}} = 6$
	$N_C = 3$	$N_{\bar{C}} = 6$	$N = 9$

(c) 'Point' feature in 'Politic' category

	$v_{Category} = v_{Politic} = 1$	$v_{Category} = v_{Politic} = 0$	
$v_{Feature} = v_{MP} = 1$	$TP = 3$	$FP = 1$	$N_F = 4$
$v_{Feature} = v_{MP} = 0$	$FN = 0$	$TN = 5$	$N_{\bar{F}} = 5$
	$N_C = 3$	$N_{\bar{C}} = 6$	$N = 9$

(d) 'MP' feature in 'Politic' category

	$v_{Category} = v_{Politic} = 1$	$v_{Category} = v_{Politic} = 0$	
$v_{Feature} = v_{Barrel} = 1$	$TP = 2$	$FP = 2$	$N_F = 4$
$v_{Feature} = v_{Barrel} = 0$	$FN = 1$	$TN = 4$	$N_{\bar{F}} = 5$
	$N_C = 3$	$N_{\bar{C}} = 6$	$N = 9$

(e) 'Barrel' feature in 'Politic' category

Table 4.14. Confusion matrix for the news example (economy category)

	$v_{Category} = v_{\text{Economy}} = 1$	$v_{Category} = v_{\text{Economy}} = 0$	
$v_{Feature} = v_{Ball} = 1$	$TP = 1$	$FP = 4$	$N_F = 5$
$v_{Feature} = v_{Ball} = 0$	$FN = 1$	$TN = 3$	$N_{\bar{F}} = 4$
	$N_C = 2$	$N_{\bar{C}} = 7$	$N = 9$

(a) 'Ball' feature in 'Economy' category

$v_{Category} = v_{\text{Economy}} = 1$	$v_{Category} = v_{\text{Economy}} = 0$

$v_{Feature} = v_{Parliament} = 1$	$TP = 1$	$FP = 3$	$N_F = 4$
$v_{Feature} = v_{Parliament} = 0$	$FN = 1$	$TN = 4$	$N_{\bar{F}} = 5$
	$N_C = 2$	$N_{\bar{C}} = 7$	$N = 9$

(b) 'Parliament' feature in 'Economy' category

	$v_{Category} = v_{\text{Economy}} = 1$	$v_{Category} = v_{\text{Economy}} = 0$	
$v_{Feature} = v_{Point} = 1$	$TP = 0$	$FP = 3$	$N_F = 3$
$v_{Feature} = v_{Point} = 0$	$FN = 2$	$TN = 4$	$N_{\bar{F}} = 6$
	$N_C = 2$	$N_{\bar{C}} = 7$	$N = 9$

(c) 'Point' feature in 'Economy' category

	$v_{Category} = v_{\text{Economy}} = 1$	$v_{Category} = v_{\text{Economy}} = 0$	
$v_{Feature} = v_{MP} = 1$	$TP = 1$	$FP = 3$	$N_F = 4$
$v_{Feature} = v_{MP} = 0$	$FN = 1$	$TN = 4$	$N_{\bar{F}} = 5$
	$N_C = 2$	$N_{\bar{C}} = 7$	$N = 9$

(d) 'MP' feature in 'Economy' category

	$v_{Category} = v_{\text{Economy}} = 1$	$v_{Category} = v_{\text{Economy}} = 0$	
$v_{Feature} = v_{Barrel} = 1$	$TP = 2$	$FP = 2$	$N_F = 4$
$v_{Feature} = v_{Barrel} = 0$	$FN = 0$	$TN = 5$	$N_{\bar{F}} = 5$
	$N_C = 2$	$N_{\bar{C}} = 7$	$N = 9$

(e) 'Barrel' feature in 'Economy' category

4.5.2. Unsupervised TW schemes

Unsupervised TW schemes do not rely on predefined information about the categories to which the training data belongs, so these schemes are useful if there is no labeled training data available. this schemes, derived from information retrieval (IR) research, has the major drawback that it does not consider the distribution of documents.

The simplest FS way is to randomly select a subset of features as needed. Since the only benefit of this type is to reduce the number of features without regard to what kind of features are chosen, we cannot expect incremental performance on each attempt, and we cannot expect performance to be constant when the number of features is increased or decreased. Each attempt will produce completely different from the one before it, and therefore it will be almost impossible to recreate the good results or to wait for better results from each attempt incrementally.

Collection frequency (CF) does not look at documents or categories in which the feature occurs. The most important is the number of frequency features in documents in general. If the feature appears in the document once, its value will be 1, if the feature appears twice, its value will be 2, and so on.

$$CF(F) = Count(F) \quad \textbf{(4.1)}$$

count(F) denotes the number of times the feature *F* is repeated in all documents and categories. Table 4.15 shows the CF of the features in the news example.

Table 4.15. News example: features ranked by CF value

Feature	CF
Ball	8.0
MP	7.0
Point	5.0
Parliament	4.0
Barrel	4.0

Document frequency (DF) is one of the simplest unsupervised TW schemes which counting the number of documents containing the feature. If we are interested in choosing the features that are repeated in many features, we can use this method and choose the features that are repeated in the greatest number of documents. DF can be calculated by (4.2),

$$DF(F) = N_F \quad \textbf{(4.2)}$$

Table 4.16 show the DF values for each feature in the news example. The results shows that this method is concerned only with the number of documents in which the feature appears, that is, a term that is repeated only once in four different documents is more important than a term that is repeated ten times in only one document. DF values are global values, meaning that we do not need to globalize the local values.

Table 4.16. News example: features ranked by DF value

Feature	DF
Ball	5.0
Parliament	4.0
Barrel	4.0
MP	4.0
Point	3.0

Term frequency- inverse document frequency (TF–1DF) is one of the most used TW schemes. TF assigns a score to each term based on the number of times the term occurs in the document, so this command may result in higher scores for common words, such as stop words, that occur frequently in documents if they are not removed earlier. The aim of the IDF is to reduce the weighting of these frequent terms, which are usually not very important terms, while giving higher weight to infrequent terms [75]. For TC applications, TF-IDF is considered a standard approach in the literature for comparing results with other TW schemes. TF-IDF is calculated as,

$$TF \cdot IDF(F, C) = \frac{f_{ij}}{\sum_{T_j} f_{ij}} \times log \frac{N}{d(t_i)} \quad \textbf{(4.3)}$$

f_{ij} denotes the number of times that feature *F* occurs in the given j^{th} document and $\frac{N}{d(t_i)}$ is the *IDF* for the feature *F*. *N* is the number of documents, and $d(t_i)$ is the collection number of documents that a feature *F* occurs at least once. In the context of TC also, due to the simple structure of the IDF, many commonly used words are often repeated but are not removed when the stop words are removed, which causes some keywords to disappear or reduce the importance of some keywords in documents [76]. Looking at the number of words in each of the nine documents we deal with in the news example, we can calculate the TF for each term as,

$$TF(Ball, \#2) = \frac{2}{4} = 0.500$$

We must find the IDF for each term before we can find the TF-IDF by multiplying the values for each term. We can calculate the IDF for each term in the following way,

$$IDF(Ball) = \log\left(\frac{9}{5}\right) = 0.255$$

In the same way, we can find the TF and IDF value for all the features shown in news example. Table 4.17 shows TF, IDF, and TF-IDF values for all features.

Table 4.17. News example: TF, IDF and TF-IDF values

TW approach	**# Document**	**Feature**				
		Ball	*Parliament*	*Point*	*MP*	*Barrel*
TF	#1	0.500	0.000	0.500	0.000	0.000
	#2	0.500	0.250	0.250	0.000	0.000
	#3	1.000	0.000	0.000	0.000	0.000
	#4	0.250	0.000	0.750	0.000	0.000
	#5	0.000	0.330	0.000	0.330	0.330
	#6	0.000	0.500	0.000	0.500	0.000
	#7	0.000	0.000	0.000	0.750	0.250
	#8	0.330	0.330	0.000	0.000	0.330
	#9	0.000	0.000	0.000	0.660	0.330
IDF		0.225	0.352	0.477	0.352	0.352
TF-IDF	#1	0.113	0.000	0.239	0.000	0.000
	#2	0.113	0.088	0.120	0.000	0.000
	#3	0.225	0.000	0.000	0.000	0.000
	#4	0.064	0.000	0.358	0.000	0.000
	#5	0.000	0.116	0.000	0.116	0.116
	#6	0.000	0.176	0.000	0.176	0.000
	#7	0.000	0.000	0.000	0.264	0.088
	#8	0.084	0.116	0.000	0.000	0.116
	#9	0.000	0.000	0.000	0.116	0.116
	Maximize	0.225	0.176	0.358	0.264	0.116

Collection frequency inverse document frequency (CF-IDF) is calculated by multiplying the value of CF and IDF as the following formula,

$$CF - IDF(F) = CF(F, C) \times log \frac{N}{d(t_i)} \quad \textbf{(4.4)}$$

We can calculate the CF-IDF feature values for the news example as follows,

$$CF - IDF(Ball) = 8 \times \log\left(\frac{9}{5}\right) = 2.042$$

In the same way, we can find the CF-IDF value for all the features shown in news example. Table 4.18 shows CF-IDF values for all features.

Table 4.18. News example: features ranked by CFIDF value

Feature	CF-IDF
MP	2.465
Ball	2.042
Point	1.908
Parliament	1.409
Barrel	1.409

4.5.3. Supervised TW schemes

Unlike unsupervised schemes, supervised TW schemes rely on category information in the training data, so pre-labeled data must be available for these schemes to perform well. Several studies have demonstrated that supervised methods provide more efficient results compared to unsupervised methods [77]. In the TC process, it is important to know the category of each document, so TC is a supervised task. This section will present several supervised TW schemes that used in this thesis.

Chi-square (CHI2) is one of the most used TW schemes in TC to test whether the occurrence of a given term and the occurrence of a given class are independent [2]. Thus, the following quantity for each term is estimated and arranged according to their scores, the CHI2 formula is expressed in (4.5).

$$CHI2(F, C) = N \times \frac{(TP \cdot TN - FP \cdot FN)^2}{N_C \cdot N_{\bar{C}} \cdot N_F \cdot N_{\bar{F}}} \quad \textbf{(4.5)}$$

A high CHI2 value indicates a higher probability of occurrence of the term and that the words tend to be keywords in the text, which will help us determine the appropriate vocabulary that will contain the most important words. To find the CHI2 value for features, the CHI2 value of the features must be found within each of the existing

categories, meaning that each feature in the news example will have three CHI2 values. First, we will calculate sport-category value as follows.

$$CHI2(Point, Sport) = \frac{9 \times \left((3 \times 5) - (0 \times 1)\right)^2}{4 \times 5 \times 3 \times 6} = \frac{2025}{360} = \frac{45}{8} = 5.625$$

In the same way, the CHI2 value is found for each feature within each category, then the features are grouped into a single global list of selected features. There are many techniques that help us to globalize the results of TW schemes, such as summing and averaging the values, and choosing the highest value for each feature. In this thesis, we will depend on the globalization of the results of TW schemes on choosing the highest value according to (4.6).

$$\begin{aligned} CHI2(F) &= max\ CHI2(F, C) \\ &= max\ CHI2(Ball, \{Sport, Politic, Economy\}) \\ &= max\ \{5.760, 5.625, 0.0321\} = 5.760 \end{aligned} \quad \textbf{(4.6)}$$

Table 4.19. News example: features ranked by global CHI2 value

Feature	Sport	Politic	Economy	Maximum
Ball	5.760	5.625	0.032	5.760
Barrel	5.760	0.900	3.214	5.760
MP	5.760	5.625	0.032	5.760
Point	5.625	2.250	1.286	5.625
Parliament	1.103	0.900	0.032	1.103

Information gain (IG) is a measure of the dependency between one random feature and one random category. IG for each feature can be calculated by (4.7).

$$IG(F) = -\sum_{k=1}^{C} \frac{N_C}{N} log \frac{N_C}{N} + \frac{N_F}{N} \sum_{k=1}^{C} \frac{TP}{N_F} log \frac{TP}{N_F} + \frac{N_{\bar{F}}}{N} \sum_{k=1}^{C} \frac{FN}{N_{\bar{F}}} log \frac{FN}{N_{\bar{F}}} \quad \textbf{(4.7)}$$

We can measure the IG value of the Ball feature in the news example as follows:

$$\begin{aligned} IG(Ball) = &-\left(\frac{4}{9} log \frac{4}{9}\right) - \left(\frac{3}{9} log \frac{3}{9}\right) - \left(\frac{2}{9} log \frac{2}{9}\right) \\ &+ \frac{5}{9} \times \left(\frac{4}{5} log \frac{4}{5} + \frac{0}{5} log \frac{0}{5} + \frac{1}{5} log \frac{1}{5}\right) \\ &+ \frac{4}{9} \times \left(\frac{0}{4} log \frac{0}{4} + \frac{3}{4} log \frac{3}{4} + \frac{1}{4} log \frac{1}{4}\right) \end{aligned}$$

$$= 0.157 + 0.159 + 0.145 + \frac{5}{9} \times (-0.078 - 0.094 - 0.151)$$
$$+ \frac{4}{9} \times (0 - 0.094 - 0.151)$$
$$= 0.461 - 0.179 - 0.109 = 0.173$$

Table 4.20 shows the global IG values for all features in the news example.

Table 4.20. News example: features ranked by IG value

Feature	IG
MP	0.231
Barrel	0.206
Ball	0.173
Point	0.168
Parliament	0.031

Mutual information (MI) measures the association between features and categories as it looks for the MI of the feature *F* in category *C*. The MI value indicates the importance of each feature in the process of correctly classifying the data. MI can be calculated by (4.8).

$$MI(F, C) = log \frac{TP \cdot N}{(TP + FP) \cdot (TP + FN)} \quad \textbf{(4.8)}$$

One of the problems with this formula is the inequality in the distribution of scores between rare and common terms, whereby higher scores are assigned to rare terms than to common terms [78]. To reduce the problem of degrees between rare and common terms, a second formula has been proposed that calculates all four possible combinations of *(F, C)* and multiplies them by their probabilities, as shown in (4.9). the globalization value of MI can be calculated by (4.10).

$$MI(F, C) = \frac{TP}{N} log \frac{N \cdot TP}{N_F \cdot N_C} + \frac{FP}{N} log \frac{N \cdot FP}{N_F \cdot N_{\bar{C}}} + \frac{FN}{N} log \frac{N.FN}{N_{\bar{F}} \cdot N_C} + \frac{TN}{N} log \frac{N.TN}{N_{\bar{F}} \cdot N_{\bar{C}}} \quad \textbf{(4.9)}$$

$$MI(F) = max\ MI(t_i, d_j) \quad \textbf{(4.10)}$$

MI values for the sport category in the news example calculated as,

$$MI(Point, Sport) = \frac{3}{9} log \frac{9 \times 3}{3 \times 6} + \frac{0}{9} log \frac{9 \times 0}{3 \times 6} + \frac{1}{9} log \frac{9 \times 1}{6 \times 4} + \frac{5}{9} log \frac{9 \times 5}{6 \times 5}$$
$$= 0.109$$

Table 4.21 shows the MI values for each feature in each category and the globalization values measured by (4.10).

Table 4.21. News example: features ranked by global MI value

Feature	Sport	Politic	Economy	Maximum
Ball	0.178	0.168	0.000	0.178
MP	0.178	0.168	0.000	0.178
Barrel	0.178	0.022	0.096	0.178
Point	0.109	0.076	0.046	0.109
Parliament	0.048	0.022	0.109	0.109

Odd ratio (OR) uses the ratio of positive and negative categories to classify documents, which is calculated using the incidence of terms, where the probabilities of a feature in one category are compared with the probability of its occurrence in another [79, 80]. OR gives higher positive scores for terms that occur frequently in one category, but rarely occur in other categories, while OR scores negative for terms that occur more frequently in other categories. If the probabilities of a given term appearing in all categories are equal, then it is given a score of zero. OR can calculate as,

$$OR(F, C) = log \frac{\left(\frac{TP}{N_C}\right)\left(1 - \frac{FP}{N_{\bar{C}}}\right)}{\left(\frac{FP}{N_{\bar{C}}}\right)\left(1 - \frac{TP}{N_C}\right)} \quad \textbf{(4.11)}$$

Equation (4.11) shows that a problem will occur if there are division by zero or the occurrence of $log(0)$ condition, so special cases that cause an error in the system are treated in the following form [7] in the thesis.

$$\frac{TP}{N_C} = \begin{cases} \frac{TP}{N_C} & if\ TP > 0 \\ \frac{1}{N^2} & if\ TP = 0 \\ 1 - \frac{1}{N^2} & if\ TP = N_C \end{cases} \quad \textbf{(4.12)}$$

OR values for the sport category in the news example calculated as,

$$OR(Ball, Sport) = log \frac{\left(\frac{TP}{N_C}\right)\left(1 - \frac{FP}{N_{\bar{C}}}\right)}{\left(\frac{FP}{N_{\bar{C}}}\right)\left(1 - \frac{TP}{N_C}\right)} = log \frac{\left(\frac{4}{4}\right)\left(1 - \frac{1}{5}\right)}{\left(\frac{1}{5}\right)\left(1 - \frac{4}{4}\right)} = Unknow!$$

The above equation is not completed due to division by 0, so it must be updated to be proportional to the special conditions in (4.12):

$$OR(Ball, Sport) = log \frac{\left(1 - \frac{1}{N^2}\right)\left(1 - \frac{1}{5}\right)}{\left(\frac{1}{5}\right)\left(1 - \left(1 - \frac{1}{N^2}\right)\right)} = log \frac{\left(1 - \frac{1}{81}\right)\left(1 - \frac{1}{5}\right)}{\left(\frac{1}{5}\right)\left(1 - \left(1 - \frac{1}{81}\right)\right)}$$

$$= log \frac{\left(\frac{80}{81}\right)\left(\frac{4}{5}\right)}{\left(\frac{1}{5}\right)\left(\frac{1}{81}\right)} = log \frac{\left(\frac{64}{81}\right)}{\left(\frac{1}{405}\right)} = 2.505$$

In a similar way, OR is calculated for all the features in each category in Table 4.20, which shows the globalization value of OR, where the largest value of the pair (feature-category) is, as shown in (4.13),

$$OR(F) = max\ OR(t_i, d_j) \quad \textbf{(4.13)}$$

Table 4.22. News example: features ranked by global OR value

Feature	Sport	Politic	Economy	Sum	Maximum
Barrel	-2.505	4.000	0.031	1.526	4.000
Parliament	-0.653	4.000	1.334	4.681	4.000
Ball	2.505	0.003	0.750	2.258	2.505
Point	2.380	0.013	0.017	2.41	2.380
MP	-2.505	0.063	1.334	-1.108	1.334

Gain ratio (GR) was introduced in 2004 by Debole and Sebastiani as the ratio of information gain to the material information in the documents we deal with [2]. GR is one of the improved methods of IG, which aims to reduce the impact of terms that are repeated in most documents, such as stop words, by giving them appropriate values for their importance [81]. GR is calculated as shown in (4.14) [82].

$$GR(F) = \frac{IG(F, C)}{\sum_{k=1}^{C} -\frac{N_C}{N} \cdot log \frac{N_C}{N} - \frac{N_{\bar{C}}}{N} \cdot log \frac{N_{\bar{C}}}{N}} \quad \textbf{(4.14)}$$

The value of the GR is between [0, 1]. GR value close to 1 indicates that the feature may be important, while this importance decreases as the GR value approaches zero [83]. We can measure the GR value of the Ball feature as follows,

$$GR(Ball) = \frac{0.173}{\begin{pmatrix} -\frac{4+0}{9} \cdot log\left(\frac{4+0}{9}\right) \\ -\frac{1+4}{9} \cdot \log\left(\frac{1+4}{9}\right) \end{pmatrix} + \begin{pmatrix} -\frac{0+3}{9} \cdot \log\left(\frac{0+3}{9}\right) \\ -\frac{5+1}{9} \cdot \log\left(\frac{5+1}{9}\right) \end{pmatrix} + \left(-\frac{1+1}{9} \cdot \log\left(\frac{1+1}{9}\right) - \frac{4+3}{9} \cdot \log\left(\frac{4+3}{9}\right)\right)}$$

$$GR(Ball) = \frac{0.173}{0.298 + 0.276 + 0.230} = \frac{0.173}{0.804} = 0.215$$

Table 4.23 shows the global GR values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.23. News example: features ranked by GR value

Feature	GR
Point	0.295
MP	0.287
Barrel	0.256
Ball	0.215
Parliament	0.039

Inverse gravity moment (IGM) is a measure of the irregularity or concentration of the terms distributed among the categories as the term should have a more concentrated distribution between the categories to have a better discrimination power for the category compared to others [84]. The IGM is calculated as,

$$IGM(F) = \frac{f_{k1}}{\sum_{r=1}^{m} f_{kr}.r} \quad \textbf{(4.15)}$$

f_{kr} is the number of documents belonging to the r^{th} category, while r is the number of categories in the data ($r = 1, 2, \ldots, M$) and f_{k1} denotes the document frequency of the feature F in category that appears most often. The IGM can be calculated by (4.16).

$$w_g(F) = 1 + \lambda \cdot IGM(F) \quad \textbf{(4.16)}$$

$w_g(t_k)$ is the IGM-based global weighting factor for the feature F and is a parameter used to maintain a balance between global and local factors in the TW. The λ parameter can have a value between 5.0 and 9.0 but it is usually set as 6.0 or 7.0 [85]. IGM values for features in the news example calculated as:

$$IGM(Ball) = 1 + 7 \times \frac{7}{(7 \times 1) + (1 \times 2) + (0 \times 3)} = 6.444$$

Table 4.24 shows the global IGM values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.24. News example: features ranked by IGM value

Feature	IGM
Point	8.000
Ball	6.444
MP	4.888
Barrel	3.333
Parliament	3.000

In Inverse category frequency (ICF), the term has greater discriminatory power the fewer categories it appears in, this discriminatory power is the force with which each term contributes to TC, and which increases the performance the higher it is. This assumption is called the ICF assumption in which the ICF favors rare terms and has biases against common terms at the category level. ICF is calculated by (4.17).

$$ICF(F) = \log\left(\frac{|C|}{cf(F)}\right) \quad \textbf{(4.17)}$$

In (4.17), let the CF denote the number of categories in which the feature *F* appears, $|C|$ Indicates the number of categories in the training set. We can measure the ICF value of the features in news example as follows,

$$ICF(MP) = \log\left(\frac{3}{2}\right) = 0.176$$

Table 4.25 shows the global ICF values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.25. News example: features ranked by ICF value

Feature	ICF
Point	0.477
MP	0.176
Barrel	0.176
Ball	0.176
Parliament	0.000

Term frequency- inverse document frequency- inverse category frequency (TF-IDF-ICF) is a supervised TW scheme that based on the traditional TF-IDF and ICF to give higher weighting values for rare terms that appear in fewer documents (IDF) and categories (ICF). TF-IDF-ICF is calculated by (4.18).

$$TF \cdot IDF \cdot ICF(F) = \frac{f_{ij}}{\sum_{T_j} f_{ij}} \times log \frac{N}{d(t_i)} \times \log \left(\frac{|C|}{cf(t_k)}\right) \quad \textbf{(4.18)}$$

The TF-IDF results shown in Table 4.17 and the ICF results shown in Table 4.25 are utilized to get the TF-IDF-ICF results shown below in Table 4.26.

Table 4.26. News example: features ranked by TF-IDF-ICF value

Feature	TF-IDF	ICF	TF-IDF-ICF
Point	0.358	0.477	0.171
MP	0.264	0.176	0.046
Ball	0.225	0.176	0.040
Barrel	0.116	0.176	0.020
Parliament	0.176	0.000	0.000

Relevance frequency (RF) improves the term discriminatory power to classify texts by focusing on the term that is frequently repeated in the positive category compared to the terms that are repeated in the negative category [86], as this method helps to choose the positive features from the negative features, and vice versa [82]. RF is calculated by (4.19).

$$RF(F, C) = \log \left(2 + \frac{TP}{\max(1, FP)}\right) \quad \textbf{(4.19)}$$

RF values for the features in news example can be found using (4.19).

$$RF(Barrel, Sport) = \log\left(2 + \frac{0}{\max(1,4)}\right) = 0.301$$

RF for all features is given in Table 4.27, which also shows the globalization value of RF, where the largest value of the pair (feature-category) is chosen, as shown in (4.20),

$$RF(F) = max\ RF(F, C) \quad \textbf{(4.20)}$$

Table 4.27. News example: features ranked by global RF value

Feature	Sport	Politic	Economy	Sum	Maximum
Ball	0.778	0.301	0.352	1.431	0.778
Point	0.699	0.301	0.301	1.301	0.699
MP	0.301	0.699	0.368	1.368	0.699
Barrel	0.301	0.477	0.477	1.255	0.477
Parliament	0.368	0.477	0.368	1.213	0.368

Inverse document frequency excluding category (IDFEC) is a variant of TF-IDF [87]. IDFEC try to avoid reducing the weights of terms that are repeated in one category,

which is the case usually caused by the standard form of IDF, as this leads to many terms, which may be important, are deprived, so in this scheme is increased weights assigned to terms that are repeated many times in the same category, and thus try to make these terms not marginalized. IDFEC is calculated as follows:

$$IDFEC(F\ , C) = \log\left(\frac{FN + TN}{\max(1, FN)}\right) \quad \textbf{(4.21)}$$

The IDFEC for the "point" feature and "sport" category in our news example using (4.21) is below.

$$IDFEC(Point, Sport) = \log\left(\frac{1+5}{\max(1,1)}\right) = 0.778$$

The IDFEC for all the features in each category according to (4.21) are calculated and given in Table 4.28. As shown in (4.22), maximum globalization has been applied.

$$IDFEC(F) = max\ IDFEC(F\ , C) \quad \textbf{(4.22)}$$

Table 4.28. News example: features ranked by global IDFEC value

Feature	Sport	Politic	Economy	Maximum
Point	0.778	0.301	0.477	0.778
MP	0.097	0.699	0.699	0.699
Barrel	0.097	0.699	0.699	0.699
Parliament	0.222	0.699	0.699	0.699
Ball	0.602	0.125	0.602	0.602

Gini Index (GI) can be briefly defined as an improved version of the trait selection algorithm used in constructing the DT [88]. It is calculated by (4.23).

$$GI(F) = \sum_{k=1}^{C} \left(\frac{TP}{N_F}\right)^2 \cdot \left(\frac{TP}{N_C}\right)^2 \quad \textbf{(4.23)}$$

We can measure the GI value of the features in news example as follows:

$$GI(Point) = \left((\frac{3}{3+0})^2 \times (\frac{3}{3+1})^2\right) + \left((\frac{0}{0+3})^2 \times (\frac{0}{0+3})^2\right) + \left((\frac{0}{0+3})^2 \times (\frac{0}{0+2})^2\right)$$
$$= 0.5625 + 0 + 0 = 0.563$$

Table 4.29 shows the global GI values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.29. News example: features ranked by GI value

Feature	GI
Ball	0.650
MP	0.566
Point	0.563
Barrel	0.361
Parliament	0.131

The initial design of the GI measured the percentage of impurities in the features, the smaller the ratio, the fewer impurities and the better the features. Improvements were added to a GI, and it was named Gini text (GT) as shown in (4.24) [88, 89].

$$GT(F) = \sum_{k=1}^{C} \left(\frac{TP}{N_F}\right)^2 \cdot \left(\frac{TP}{N_C}\right)^2 \cdot \left(\frac{FN}{N_{\bar{F}}}\right)^2 \cdot \left(\frac{FN}{N_{\bar{C}}}\right)^2 \quad \textbf{(4.24)}$$

We can measure the GT value of the features in news example as follows,

$$\begin{aligned} GT(Parliament) = &\left(\left(\frac{1}{1+3}\right)^2 \times \left(\frac{1}{1+3}\right)^2 \times \left(\frac{3}{3+2}\right)^2 \times \left(\frac{3}{3+2}\right)^2\right) \\ &+ \left(\left(\frac{2}{2+2}\right)^2 \times \left(\frac{2}{2+1}\right)^2 \times \left(\frac{1}{1+4}\right)^2 \times \left(\frac{1}{2+4}\right)^2\right) \\ &+ \left(\left(\frac{1}{1+3}\right)^2 \times \left(\frac{1}{1+1}\right)^2 \times \left(\frac{1}{1+4}\right)^2 \times \left(\frac{1}{3+4}\right)^2\right) \\ &= 5.063 \times 10^{-4} + 1.235 \times 10^{-4} + 1.276 \times 10^{-5} \\ &= 6.426 \times 10^{-4} \end{aligned}$$

Table 4.30 shows the global GT values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.30. News example: features ranked by GT value

Feature	GT
Parliament	6.426×10^{-4}
Point	6.250×10^{-4}
Barrel	1.235×10^{-4}
MP	1.276×10^{-5}
Ball	1.276×10^{-5}

Distinguishing feature selection (DFS) is one of the most recent TW schemes and is also a global measure of FS [90]. DFS select distinct and important features while reducing the impact of marginal features by normalizing the weight of the terms

considering some predefined criteria. This method gives a weight to the terms in the range [0, 1] determined as,

$$DFS(F) = \frac{\frac{TP}{N_F}}{\frac{FN}{N_C} + \frac{FP}{N_{\bar{C}}} + 1} \quad \textbf{(4.25)}$$

where $\frac{TP}{N_F}$ is the conditional probability of category *C* given presence of feature *F*, $\frac{FN}{N_C}$ is the conditional probability of absence of feature *F* given category *C*, and $\frac{FP}{N_{\bar{C}}}$ is the conditional probability of feature *F* given all the categories except *C*. We can measure the DFS value of the features in news example as follows,

$$DFS(Parliament) = \left(\frac{\frac{1}{(1+3)}}{\frac{3}{(3+1)} + \frac{3}{(2+3)} + 1}\right) + \left(\frac{\frac{2}{(2+2)}}{\frac{1}{(2+1)} + \frac{2}{(2+4)} + 1}\right) + \left(\frac{\frac{1}{(1+3)}}{\frac{1}{(1+1)} + \frac{3}{(3+4)} + 1}\right) = 0.536$$

Table 4.31 shows the global DFS values for all features in the news example, which calculated in the same way as for the previous feature.

Table 4.31. News example: features ranked by DFS value

Feature	DFS
Point	0.800
MP	0.773
Ball	0.763
Barrel	0.662
Parliament	0.536

Galavotti-Sebastiani-Simi (GSS) coefficient was originally proposed as a numerator of the CHI2 function [91]. Later it was called GSS coefficient in relation to the names of the three inventors [63]. GSS is calculated as follows,

$$GSS(F, C) = (TP \cdot TN) - (FN \cdot FP) \quad \textbf{(4.26)}$$

The difference is clear between the CHI2 equation and the GSS equation, where the *N* factor for the number of documents and the denominator has been completely removed. The denominator has been completely removed because it gives a high

correlation coefficient for rare words and rare categories [92]. In this method, experiments have shown that the results are much better using maximum as a globalization strategy instead of the average [93]. The globalization value of GSS as,

$$GSS(F) = \max_{k=1} GSS(F, C) \tag{4.27}$$

GSS values for Ball feature in all categories in the news example can be calculated as,

$$GSS(Ball, Sport) = (4 \times 4) - (0 \times 1) = 16 - 0 = 16$$
$$GSS(Ball, Politic) = (0 \times 1) - (3 \times 5) = 0 - 15 = -15$$
$$GSS(Ball, Economy) = (1 \times 3) - (1 \times 4) = 3 - 4 = -1$$

$$GSS(Ball) = \max_{k=1} GSS(Ball, \{Sport, Politic, Economy\})$$
$$= \max\{16, -15, -1\} = 16$$

The rest is given in Table 4.32.

Table 4.32. News example: features ranked by GSS value

Feature	Sport	Politic	Economy	Maximum
Ball	16	-15	1	16
Point	15	-9	-6	15
MP	-16	15	1	15
Barrel	-16	6	8	8
Parliament	-7	6	1	6

Ng-Goh-Low (NGL) coefficient is a is a variant of the CHI2 [94]. It was also called correlation coefficient (CC), but NGL in relation to the names of the three inventors Ng, Goh and Low [63] is preferred in this thesis. The value of NGL can be calculated as,

$$NGL(F, C) = \frac{\sqrt{N} \cdot \big((TP \cdot TN) - (FN \cdot FP)\big)}{\sqrt{N_C \cdot N_{\bar{C}} \cdot N_F \cdot N_{\bar{F}}}} \tag{4.28}$$

A positive NGL value indicates that the term is an important and distinct word associated with class c_i, while a negative value indicates that the word is probably not highly significant and is associated with class $\bar{c_i}$. The results of NGL in many studies were better than the results of the CHI2, it was also better than the OR and MI at some feature set sizes [94, 95]. NGL values for the features in our news example can be found using (4.28).

$$NGL(Parliament, Sport) = \frac{\sqrt{9} \cdot [(1 \times 2) - (3 \times 3)]}{\sqrt{4 \times 5 \times 4 \times 5}} = \frac{-21}{20} = -1.050$$

The NGL for all the features in each category according to (4.28) are calculated and given in Table 4.33. As shown in (4.29), maximum globalization has been applied.

$$NGL(F) = max\ NGL(F\ , C) \quad \textbf{(4.29)}$$

Table 4.33. News example: features ranked by global NGL value

Feature	Sport	Politic	Economy	Maximum
Ball	2.400	-2.372	0.179	2.400
Point	2.372	-1.500	-1.134	2.372
MP	-2.400	2.372	0.179	2.372
Barrel	-2.400	0.949	1.434	1.434
Parliament	-1.050	0.949	0.179	0.979

Categorical proportional difference (CPD) is one of the latest TW schemes, a novelty among the TW schemes that we will use [96]. This scheme uses only positive examples from training data. This technique measures the degree to which a particular feature distinguishes one category from another. Equation (4.30) shows how to calculate the CPD for a feature, while (4.31) the globalization CPD values can calculated by (4.31).

$$CPD(F\ , C) = \frac{TP - FP}{N_F} \quad \textbf{(4.30)}$$

$$CPD(F) = max_k\{CPD(F\ , C)\} \quad \textbf{(4.31)}$$

CPD values for the features in the news example can be found using (4.30) and (4.31),

$$CPD(Ball, Sport) = \frac{4 - 1}{5} = 0.6$$

$$CPD(Ball, Politic) = \frac{0 - 5}{5} = -1.0$$

$$CPD(Ball, Economy) = \frac{1 - 4}{5} = -0.6$$

$$CPD(Ball) = \max\{0.6, -1.0, -0.6\} = 0.6$$

In a similar way, we will calculate the CPD for all the features according to (4.30) and (4.31) and the values shown in Table 4.33.

Table 4.33. News example: features ranked by global CPD value

Feature	Sport	Politic	Economy	Maximum
Point	1.000	-1.000	-1.000	1.000
Ball	0.600	-1.000	-0.600	0.600
MP	-1.000	0.500	-0.500	0.500
Parliament	-0.500	0.000	-0.500	0.000
Barrel	-1.000	0.000	0.000	0.000

4.5.4. Other TW schemes

Despite the large number of TW schemes, supervised and unsupervised, that have been mentioned in this section, and that will be used in this thesis, which may be the largest number of terminology weighting schemes that were used in one Arabic TC research at all, there are many other TW schemes that have not been used due to time constraints. This section provides a brief overview of some of the schemes that were not used in this thesis.

Probabilistic inverse document frequency (PIDF) is a developed TW scheme for the IDF method [97]. In this method, the IDF factor is replaced by the PIDF which can calculate as,

$$PIDF(F, C) = log \frac{N - d(t_i)}{d(t_i)} \quad \textbf{(4.32)}$$

Note from the above equation that PIDF differs from IDF in that it uses the number of unrelated documents instead of the total number of documents in the numerator [98].

Weighted inverse document frequency (WIDF) has been proposed because TF-IDF scheme suffers from a clear problem, which is that we must recalculate the TF-IDF value for all documents when adding a new document, because TF-IDF depends entirely on the number of documents [99]. WIDF can be calculated using (4.33),

$$WIDF(F, C) = \frac{1}{\sum_{d_m \in D} TF(F, d_m)} \quad \textbf{(4.33)}$$

WIDF overcame the problem of TF-IDF by weighting the terms that summarize one over the set of texts. Bi-Normal separation (BNS) was first proposed in 2003 to combine with SVM classifier [100]. In 2008, BNS was developed to scale the feature values magnitude. The standard BNS shown in (4.35).

$$BNS(F,C) = \left| F^{-1}\left(\frac{TP}{N_C}\right) - F^{-1}\left(\frac{FP}{N_{\bar{C}}}\right) \right| \quad \textbf{(4.35)}$$

where F^{-1} is the standard normal distribution's inverse cumulative probability function. Since the value of the function F^{-1} will explode if $\frac{TP}{N_C} = 0$ or $\frac{FP}{N_{\bar{C}}} = 0$, these values are usually set at a minimum rate of *5/10000* to prevent the function from bursting. When setting the minimum value, we automatically set the maximum as well, which is *1 - 5/10000*. BNS finds the local value of the feature within the category, so we must use (4.36) to find the global value of each feature.

$$BNS(F) = max\ BNS(F\ ,C) \quad \textbf{(4.36)}$$

Probability-based (TF-Prob) is a supervised probability-based TW scheme. This TW scheme combines TP/FN ratios that measure the suitability of the term F to the C category, with this ratio tending to be higher if the term F is more closely related to the c_i category than the other categories. The strength of the term can be calculated using TP/FN and TP/FP, which are related indicators, using the following equation,

$$Prob - based = \log\left(1 + \frac{TP}{FP} \cdot \frac{TP}{FN}\right) \quad \textbf{(4.37)}$$

4.6. FS

FS is one of the most important TC steps, as it saves and reduce the computational cost, which positively affects the performance. In FS, the focus is placed on the salient features of the texts and the selection of features that refer to salient information that determine the general shape of the text [100]. FS helps reduce the problem of over-allocation and reduce the time required to find the best results [63]. All FS methods share have main reasons for choosing some features over others, which are [93],

1- Accuracy: Several studies have shown that results are better using a small, carefully selected set of features rather than many features, most of which will be unnecessary features that do not add meaning to the classification process.

2- Scalability: The classification process requires high-specification devices to obtain good results, so choosing a small, carefully selected group of features usually leads to significant time savings.

4.6.1. FS vs. construction and extraction

Big number of features can be used TC, but it will be useless, as the quality of the results will be low, and it will take a long time to finish. If we consider the model as the engine, the features can be considered as the fuel needed to run this engine, as we must choose the features carefully if we want to get good results. The dimensions of the originals are very large, so we must use a method to reduce these dimensions. There are many ways to reduce these large dimensions, these methods are sometimes called 'reducing dimensions methods' [65, 79], The most popular feature reduction methods are FS, feature construction and feature extraction. Figure 4.2 shows the difference in the way FS and Feature Extraction works. In FS (also called feature space reduction [63]), the number of features is greatly reduced by selecting and retaining the best *n* features obtained using TW schemes and ignoring other features. Feature construction and feature extraction transform a set of available features into fewer features by reformatting, combining, and transforming basic features into new features that are not at all like the original features.

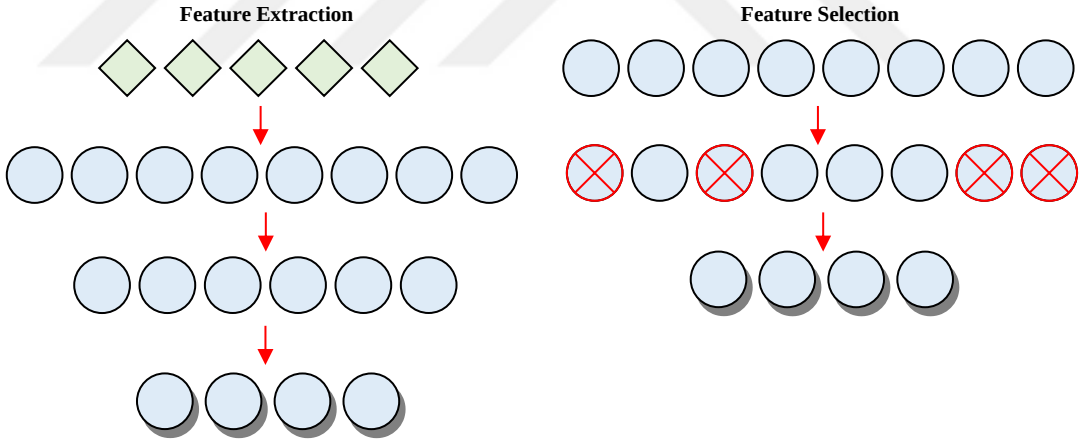

Figure 4.2. The difference between the FS and feature extraction

4.6.2. Local vs. global FS

If the number of features is few, an ordered list of features for use in training a classifier is usually made. This category list is called a local class list since the feature list is specific to only one category. Dealing with one set of data belonging to one category is often impractical for many reasons, including that the time required for classification will be much more, and accuracy is usually variable, so researchers sometimes want to combine ordered lists in one common feature list to be used with all categories. This

common list is called a global feature list, which consists of collecting all the local feature lists into a single list. The TC performance is relatively lower with the global features, as the features are less specific to categories, but it is a common type in the literature for many reasons, including that its performance is not much less than the selection of local features in return saves a lot of time, especially when dealing with big data that has a lot of features, so global FS will be used in this thesis. Table 4.34 shows an example of the difference between local and global FS approaches.

Table 4.34. Local and global FS for multilabel classification

Rank	Global		Local (Economy)		Local (Sport)		Local (Medical)	
	word	translate	word	translate	word	translate	word	translate
1	القدم	Foot	برميل	barrel	القدم	Foot	اكتشاف	invention
2	كرة	Ball	أسهم	stock	كرة	Ball	جائحة	pandemic
3	برميل	barrel	قيمة	share	ملعب	Stadium	علاج	medication
4	اكتشاف	invention	اقتصاد	Economy	نقطة	Point	طبيب	doctor
5	أسهم	stock	تزويد	supply	تدريب	training	فحوصات	Examinations

4.6.3. Filter and wrapper approach

FS methods are often grouped into filter and wrapper, as Figure 4.3 shows. Filter FS methods measure the suitability of the best set of features selected based on their scores in the TW schemes. Wrapper methods measure the suitability of sub-features using AI search methods such as the hill-climbing algorithm, the simulated annealing algorithm, and the GA, which is used in this thesis. In contrast to filtering methods, wrapper FS methods check all features over and over and change the features on each attempt which in theory avoids redundant features and leads to better results [65]. The filter FS methods are often much faster than the wrapper FS methods, especially when using large documents with many features, as the wrapper FS methods is computationally expensive due to constantly changing the features that are tested on each attempt. The model may be more susceptible to overfitting when using the wrapper FS method than using the filter FS methods. While filter FS methods may fail to find the best subset of features, because they only count on a specific number of features based on the scores of the TW schemes, the wrapper FS methods ensures that the best subset of features is obtained if appropriate parameters are chosen.

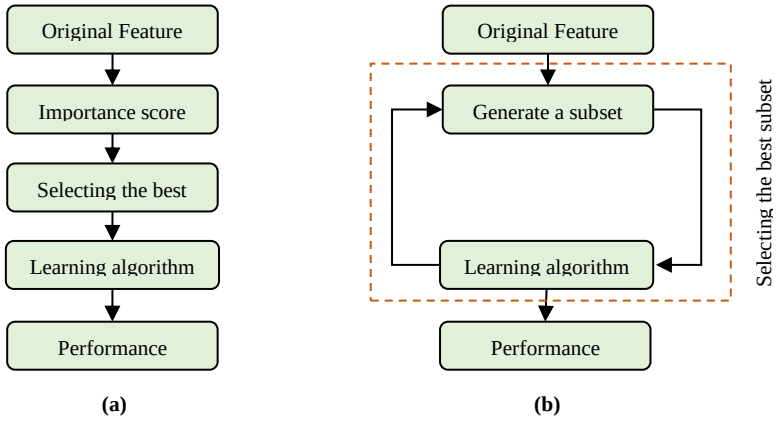

Figure 4.3. The difference between (a) filter and (b) wrapper methods

4.7. Text Classifiers

TC techniques are divided into two main parts, the statistical and ML approach as shown in Figure 4.6. Statistical techniques depend on fixed assumptions and equations developed by experts and engineers of statistical techniques, so the need for ML techniques is very little, in contrast ML techniques were invented specifically for automation [101]. Figure 4.4 shows the types of TC techniques.

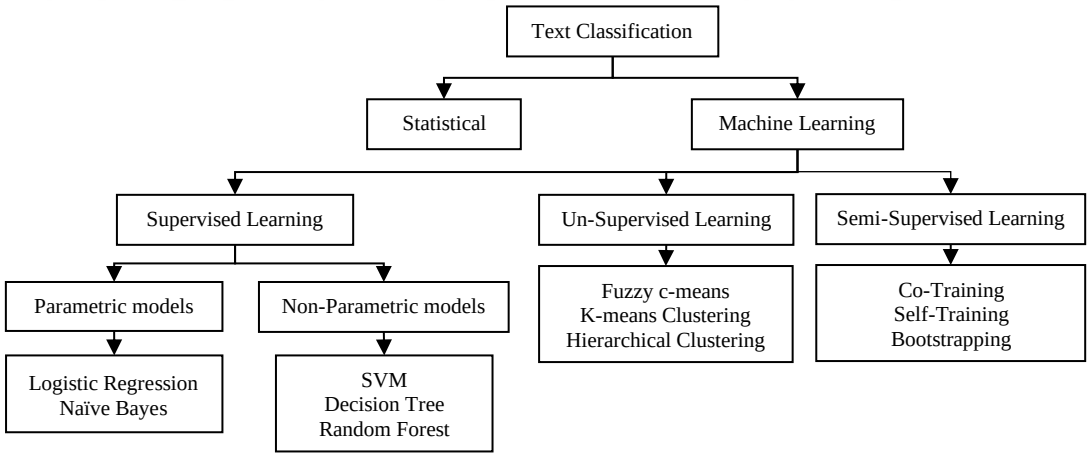

Figure 4.4. Various TC techniques

4.7.1. Statistical approach

Statistical techniques are purely mathematical operations that are introduced by engineers and experts and serve as a mathematical basis for all other text compilations. These techniques implement the instructions provided to them and give specific results according to the mathematical equation that was entered without any special ability of

their own. Statistical techniques are better in dealing with summary data, so the dimensions of large data must be reduced before using statistical techniques in TC [102]. Several statistical feature extraction techniques such as principal component analysis and mean neighbor margin maximization have proven to be good techniques for dimensionality reduction in linear data, but they are not suitable for use with non-linear data and large datasets [103].

4.7.2. ML approach

With the large increase in the number of text data that must be dealt with in all areas of life, there is a great need for automated classification systems that can automatically classify texts into predefined categories. TC using ML can be divided into three categories: supervised, unsupervised, and semi-supervised TC.

Supervised learning is the most common type because the accuracy of its results is often higher than the other two types, despite some of the difficulties that this type involves. Supervised learning requires direct human intervention before the classification process is implemented by assigning labels to all documents, which can be very difficult, especially in big data. Another difficulty is the long time it can take to get results, even though these techniques often use AI simulation. Supervised learning can also be divided into two approaches, parametric and non-parametric. Figure 4.5 shows how the supervised TC techniques work.

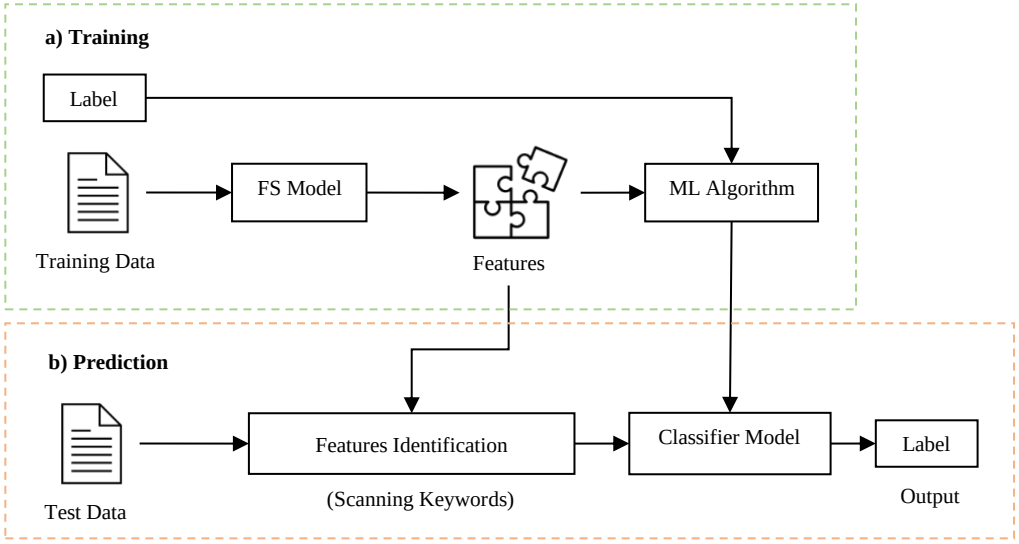

Figure 4.5. Supervised TC techniques framework

Parametric ML algorithms are the algorithms that simplify a function into a known understandable form [104]. In other words, the parametric model summarizes the training data, regardless of its number, into a set of parameters of a fixed size, where a change in the number of training data do not change its opinion about how many parameters it needs. This simplicity in understanding and interpreting results is one of the most important features of parametric machine learning algorithms, as this feature gives them speed in obtaining results and the ability to adapt, as these models can provide very good results even if the number of training data is few or if the training data is not ideal to a large extent. The simplicity may constitute a weakness in some cases, as this specific functional form and its commitment to not complicating the problem to a large extent may make it unsuitable for dealing with large problems and more suitable for simpler problems. Two of the most popular parametric ML algorithms, logistic regression (LR) and naive bayes (NB), are given below.

LR is one of the most popular supervised classification algorithms because of its flexibility in dealing with different problems. LR is used to categorize documents based on the logistic function [105], this model recognizes a vector of variables and evaluates the weights for each input variable and then predicts the category of the document as a word vector [106]. LR is superior to the simpler type of regression, linear regression, which not fit all the data points that are dealt with, while LR can cover all or most of the data due to its elastic nature. Figure 4.6 shows an example of the slope we get using LR.

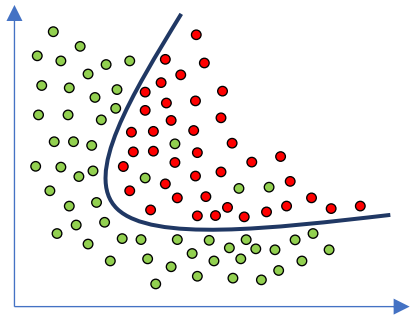

Figure 4.6. Sigmoid function used in LR

Looking mathematically at a LR function that estimates a multiple linear function is defined as,

$$LR\,(S) = b_0 + b_1 M_1 + b_2 M_2 + b_3 M_3 + \cdots + b_k M_k \qquad \textbf{(4.38)}$$

In (4.38), *S* refers to the probability of presence of feature of interest, M_k refers to the predictor value, and *b* refers to the intercept of the model.

NB is a popular classification algorithm that performs classification based on the probabilities of events. NB is mainly based on bayes theorem which assumes that there is no correlation between the variables, where we express each variable by itself and then these features are referred to as naïve. Since it is considered a naive feature set, NB algorithm can be trained using several data less than the number needed to train many other classification algorithms. Bayes theorem is based on the following formula,

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \qquad \textbf{(4.39)}$$

where P(A|B) is a probability of A given evidence B has already occurred, P(B|A) is a probability of B given evidence A has already occurred, P(A) and P(B) is a probability that A and B will occur, respectively. In this thesis, we used Bernoulli NB that have proven very effective in classifying texts [107].

Bernoulli NB is one of the highly effective classifiers in classifying texts, especially short texts, as this method is characterized by high speed in obtaining results and high quality in dealing with multiclass data, especially when the number of data is not very large [108]. In Bernoulli NB, each document is represented by a binary feature vector where the features are independent binary variables indicating whether words occur in the document or not. This model does not care how many times a word appears in the document, so, in cases where calculating word frequency is less important, Bernoulli may give better results. The Bernoulli NB decision rule is based on,

$$P(x_i|y) = P(i|y)x_i + \big(1 - P(i|y)\big)(1 - x_i) \qquad \textbf{(4.40)}$$

According to the decision rule formula shown in (4.40), x_i needs to be binary.

Non-parametric ML algorithms are the algorithms that do not summarize data and does not make strict assumptions that determine the shape of the assignment function. Thus, the non-parametric model is completely free to learn any functional form of the

training data, so this type is very good when we have big data with no prior knowledge of the data. Since this type requires many training data to obtain high-accuracy results, these algorithms may suffer from a longer time to obtain results, which may expose the model to overfitting. We will explain two of the most popular non-parametric ML algorithms, linear support vector machine classifier (Linear SVC) and random forest algorithm (RF).

SVM was first introduced by Vapnik [109], and later developed for scripting purposes by Joachims [110]. SVM is one of the most popular classification algorithms that was originally developed for classification of binary data, but it performs very well with multi-class data as well. In this algorithm, a model is built to predict the appropriate category for each document by separating the training data into two categories and the support vectors, which are chosen as only active elements in the training set, are relied upon to choose the appropriate category for each document [111, 112]. Figure 4.7 shows the linear and non-linear classifier used for 2D datasets where the red dots are class 1 data, the blue dots are class 2 data, and the yellow dots are unclassified data.

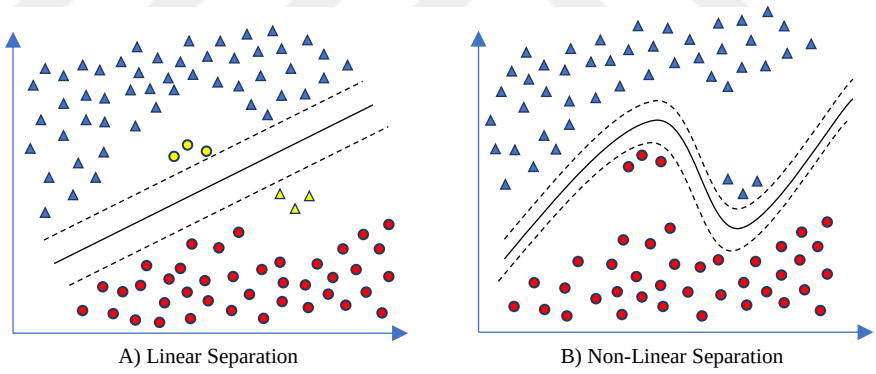

Figure 4.7. Linear and non-linear SVM

SVM is excellent at solving many predictive problems in real-world data mining such as time-series prediction, TC [113]. One of the most popular SVM algorithms is linear support vector classification, which has several features that make it highly suitable for applications with a large number of samples such as choice flexibility and loss functionality [114].

RF Algorithm is closely related to the DT algorithm. The working principle of the DT depends mainly on the classification of training data into groups. These trees consist mainly of nodes and branches. Each node represents features in the class being

classified and each subset defines a value that a node can take [115, 116]. Figure 4.8 shows the structure of the DT algorithm [117]. The DT is optimal when all training data are on the same node from the tree, but this case rarely occurs due to the presence of many invalid data, random data [118, 119].

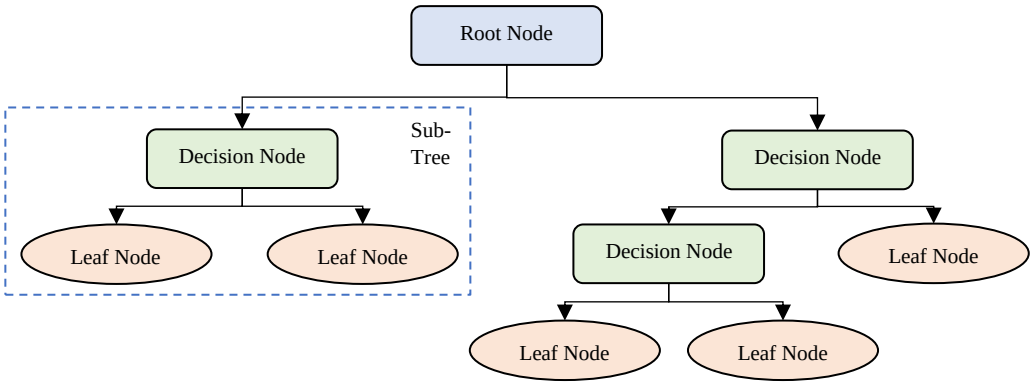

Figure 4.8. The structure DT algorithm

In RF algorithm, a very large number of DTs are created, As shown in Figure 4.9. We can define this algorithm more precisely as a set of DTs that are created according to a certain number in the pre-processing step [120].

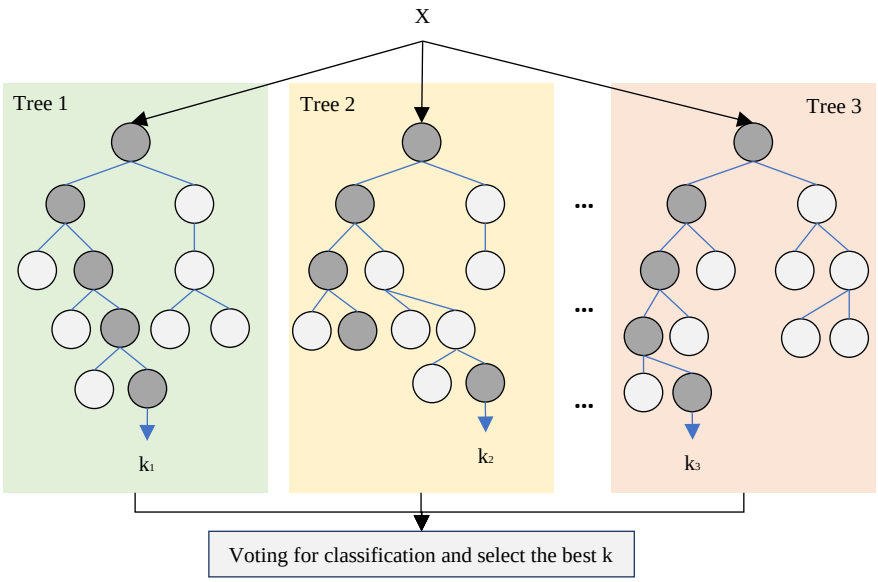

Figure 4.9. RF algorithm structure

Although the RF algorithm has many advantages such as its great ability, high flexibility in dealing with many input variables without deleting any of them, obtaining results with high efficiency, and save the generated trees or forests for future use, but

it is also is not without opportunities for error, as it is possible that there is a correlation between two trees in the forest, which leads to an increase in the error rate, and that each tree in the forest has its strength, that is, the tree with the high error rate is also powerful even though the RF algorithm ensures that a tree with a low error rate is a strong classifier and vice versa. The RF mathematical formula is,

$$n_{ij} = w_j C_j - w_{left(j)} C_{lest(j)} - w_{right(j)} C_{right(j)} \quad \textbf{(4.41)}$$

where n_{ij} indicates the significance of the j node, w_j the weighted number of samples arriving at the j node, C_j the impurity value of the j node, while the $left(j)$ and $right(j)$ refer to the sub-node of the left and right split on the j node.

In unsupervised learning, no training data is provided to the system, so the system extracts information from the data by collecting data in different groups without the need for prior information about the data categories. The model suffers from some difficulties in the beginning, when the number of data is small, because the system cannot extract enough information from the data, but when more data is entered, we will notice that the quality of the results of unsupervised algorithms increases significantly, so this type of forms are more suitable for high volume data. If the available data are not sufficient to label samples, one of the famous unsupervised learning techniques such as correlation coefficient is used to group the documents and use it to label the documents [121]. Some of the famous unsupervised learning algorithms are expectation maximization algorithm and principal component analysis [122].

In semi-supervised learning, supervised and unsupervised learning techniques are used [123]. This type uses a set of data containing a small amount of labeled data and a large amount of unlabeled data for training. The system works by combining labeled and unlabeled data, whereby unlabeled data reduces the impact of a small number of labeled data, which can negatively affect the performance. It is very important in semi-supervised learning techniques that the named data, even if few, include all categories in the data, since traditional TC approaches become empty when there is no data labeled for a particular class of dataset, so it is recommended to use a semi-supervised algorithm. Some of the SSL techniques include self-training and bootstrapping [124].

4.8. Genetic Algorithm

GA, which created by john Holland [125, 126], is one of the evolutionary algorithms which derives its way of working from the theory of evolution founded by Charles Darwin [127]. It is an easy-to-implement algorithm with a high ability to deal with complex problems and fixed and continuous data [128]. GA contains evolution parameters such as selection, crossover, and mutations, where the algorithm aims to solve all optimization problems[3] using a method like the method of natural evolution. Since this algorithm deals with individuals like real life in which each individual takes his own direction, GA has the ability to search within very large space at the same time and with remarkable efficiency. The first step of the GA in solving any problem is to randomly generate the initial population, where each solution in the search space is called a chromosome, as shown in Figure 4.10. Each chromosome contains a set of genes, which make up the structure of that chromosome. The chromosomes available in each generation are called the population.

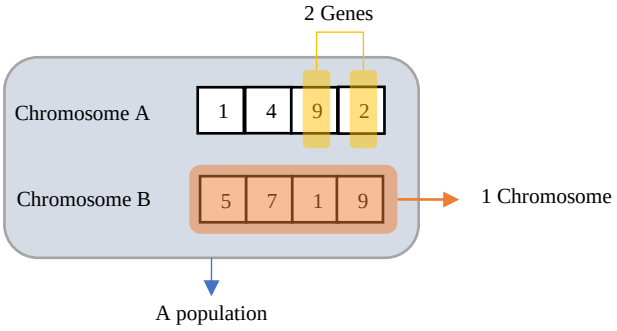

Figure 4.10. Chromosome representation

The initial set of individuals contains its own characteristics that can be changed and modified with the succession of generations. One of the most important parameters that must be carefully selected when dealing with a GA is to determine the representation that we will use to represent our solutions, as inaccurate and inappropriate representation of the problem we are dealing with can lead to poor performance of the algorithm in general. Therefore, choosing the appropriate

[3] Optimization problems are computational problems in which the aim is to find the best possible solutions.

representation will greatly contribute to the success of the GA and obtaining high-accuracy results. Figure 4.11 shows three of the most used data types used with GA.

Binary numbers representation	1	0	1	1	0	0	1	0
Integer numbers representation	5	2	3	4	6	7	1	8
Real numbers representation	1.2	9.4	3.7	4.4	5.1	0.6	9.2	6.3

Figure 4.11. Data representation type

After the initial population, or population in each generation, is established, the fitness value of each individual (chromosome) is measured, and the best individuals are selected to be the parents who will produce the offspring of the next generation with more robust traits. As these steps are repeated, GA gets closer to finding the optimal solution where the good parents selected in each generation produce better offspring. GA stops when one of the predefined stopping conditions is met, such as reaching the maximum number of generations or reaching an optimal value. GA is one of the distinguished algorithms that can be applied to solve many unconventional problems, as it approaches the optimal solution, although we may need different operations because GA is a probabilistic algorithm, that is, it produces different solutions in each operation to reach the ideal solution. Figure 4.12 shows the scheme of the GA [129].

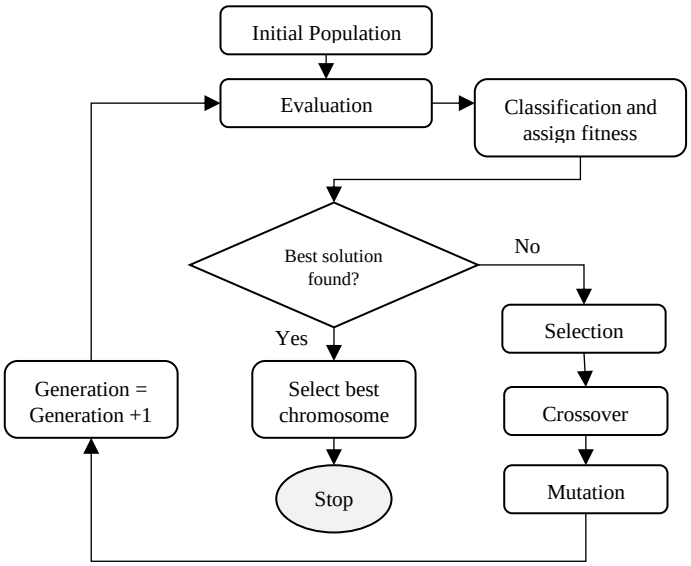

Figure 4.12. GA working principle

4.8.1. GA for FS

Because the GA is insensitive to noise and needs less knowledge about the problem it is working with, it can deal very effectively with dimensionality reduction and FS in TC techniques. Using the GA as an approach to FS, GA follows some fixed steps which, as mentioned, derive its main idea from biological processes.

The first step in the GA is to randomly generate the values of the initial population within the specified range and then the features in each chromosome are encoded as binary vectors [130], where the values used are only 0 and 1, the bit with value 1 means that the corresponding feature has been selected and 0 means the feature is unselected. Once we have created the initial set and converted the values into binary form, we must assign fitness values to determine how good or bad the members of the resulting generation are. Individuals will be selected based on their scores using the evaluation metric that will be used (f1-score).

The selection operator replicates good results and eliminates or reduces bad solutions. There are different ways to select the results, Table 4.35 shows the method that will be used in this thesis, and it is the most common selection method, the roulette wheel method, in which the parts on the roulette wheel are divided for each individual depending on the fitness value of this individual, then the individuals are chosen randomly, the probability of selecting of each individual calculated using (4.42) where the probability of selection the best individuals is higher than the rest.

$$P(F) = \frac{F1 - score(F)}{\sum_{i=1}^{n} F1 - score(F)} \quad \textbf{(4.42)}$$

This method does not guarantee that the best individuals are selected every time, but it may usually be the most appropriate method. There are many other selection methods that work in different ways such as tournament selection, rank selection and Boltzmann selection [131].

Table 4.35. Roulette wheel selection method

Chromosome	Fitness (F1 score)	Probability of choice	Selected individuals
1	0.902	23.4%	◀
2	0.882	22.9%	
3	0.831	21.6%	◀
4	0.711	18.4%	
5	0.529	13.7%	

Selected individuals (parents) are transferred to the crossover stage. In this step, new individuals will be created. The crossover rate is usually high, it is recommended that the ratio be higher than 70% to ensure good quality results [132]. The most common type of crossover is single point crossover where a specific point is determined at which the chromosome of both parents is divided into two parts and then the genes are exchanged between the parents [133, 134]. The crossover point is chosen randomly, as this point should divide the chromosome into two parts, and then the parts of the chromosomes to the right of the crossing point are then exchanged to produce two offspring [135]. An example of a single point crossover is shown in Figure 4.13.

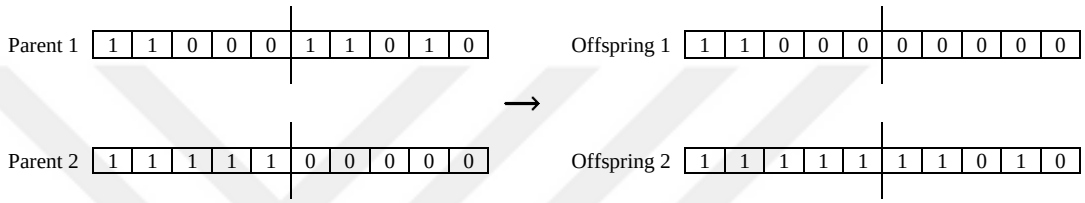

Figure 4.13. A single-point crossover operator

The mutation operator aims to create new solutions, increase the diversity of the population, and provide a way out of the local optimum by creating unique new chromosomes different from the first generation (parents) [136]. The mutation occurs at a very low probability, ranging from 0.005 to 0.1 [133, 137], as they work by shifting one of the bits of the children from 0 to 1 or vice versa as shown in Figure 4.14.

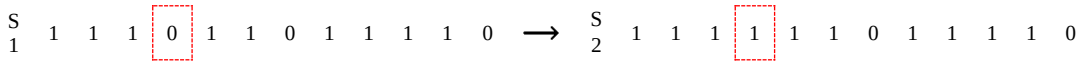

Figure 4.14. Mutation operator (S1 is the old solution and S2 is the new solution.)

After the end of all the previous steps, we reach the end of the algorithm, we now need to define the stopping criteria. There are many stopping criteria that are used in studies, the most common stopping criteria is reaching the maximum number of generations specified in the parameters, repeating the same results for a certain number of generations, and reaching a certain value that represents the optimal value [138].

4.8.2. GA parameters

The parameters are one of the most important elements that must be chosen very carefully, as any error in the criteria will greatly affect the performance and may lead to illogical results [139]. We can change parameters constantly to get the best possible

results. GA, as in all other algorithms, has a set of important parameters, including, the number of generations, population size, crossover probability, mutation rate, and elitism size, which are given in the following paragraphs.

The number of generations - the number of generations is related to the number of times the algorithms repeat all the steps of the work, so we will have a greater chance to improve the quality and get better results in each generation. Usually, the generations show a great improvement in the first generations and then show a slight progress in the later generations due to reaching a level close to the optimum level.

Population size - Population refers to the number of chromosomes present in a population (in one generation), as shown in Figure 4.15. If there are too few chromosomes, and therefore too few populations, the improvement in the results will be very slow because GA has few possibilities to make the exchange and only a small part of the search space is explored. Likewise, if there are too many chromosomes, and therefore the population is too large, GA will be much slower, causing suboptimal results. We must try many values of the population size and choose the appropriate value in which the population size is not too small or too large, which in both cases negatively affects the results of the algorithm.

Crossover probability - The crossover probability indicates the number of times the crossover process will be performed between parents to produce offspring. If there is no crossover, the resulting generations will be a clone of the parents, and at 100% probability of crossover, all offspring are by cross. Therefore, we must determine the possibility of an appropriate intersection, to search for the next generation to be new children resulting from the intersection of the parents who were found in the previous stage and some of the chromosomes of the parents that have not changed. The possibility of crossing over is very important because the process of crossover is performed in the hope that the new chromosomes will contain good parts of the old chromosomes and perhaps the new chromosomes will be better.

Mutation rate - Mutation rate shows how many times parts of a chromosome will mutate. Usually, the value of the mutation is very low, but it must be present to prevent GA from falling into local extremes. If there is no mutation rate in the algorithm, the offspring will be taken after the crossing process without any change in the chromosomes, which will lose the GA the diversity advantage necessary to obtain the

ideal results, especially in the advanced generations. If the mutation rate is too high, the chromosome will change to a completely different chromosome, which means that GA has switched to a completely random search. An appropriate mutation rate is usually selected within the range commonly used in research, less than 0.1.

Elitism size - Many good chromosomes are found in each generation of which many may be lost after the creation of a new population by crossover and mutation, from this principle the idea of elitism was introduced. In elitism, the best chromosome, or the best certain number of chromosomes, is passed on to the next generation directly, without going through the processes of crossovers and mutations that the rest of the chromosomes will undergo. Elitist can increase the quality of solutions in each generation, as well as ensure that, at the very least, we will not lose the best chromosomes we found in the previous generation, as Figure 4.15 shows.

Chromosome	Fitness
P1	0.713
P2	**0.882**
P3	0.829
P4	0.650
P5	**0.867**
P6	0.801

Current generation

→

Chromosome	Fitness
P1'	**0.882**
P2'	**0.867**
P3'	0.903
P4'	0.851
P5'	0.812
P6'	0.799

Next generation

Figure 4.15. Elite selection strategy from each generation

4.9. Training, Test, And Validation Sets

In this thesis, supervised FS methods and supervised TC techniques will be used, so working with test data and training data should be one of the most important steps to focus on. There are many techniques that have been studied for selecting test data and training data.

The first method is the training and test method (also called hold out method) and a common, and perhaps the most widely used method, is to split the available data into two sets, a training set, and a test set. The data is usually divided unevenly as two-thirds of the available data is used for training, and the remaining one-third is used for testing [24]. The training set is used to train the classifier so that it can handle and predict the results well when tested using the test data used to test the classifier and evaluate the effectiveness of its results. The results of the training data not only assess

the accuracy of the data, but also the accuracy of the TW scheme and the performance of the classifier as well. Figure 4.16 shows how the hold out method works.

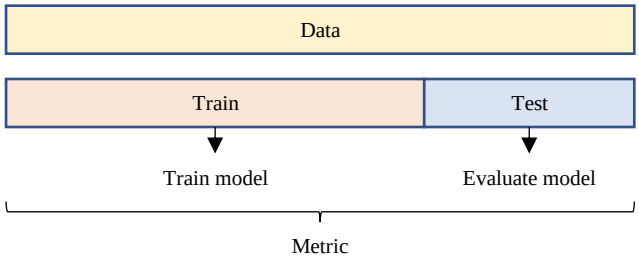

Figure 4.16. Hold out method for training a ML model

Another method for evaluating classifiers and dividing the data into test data and training data is called *k*-fold cross-validation [140], in which the documents are divided into *k* sets of equal size, *k*-1 of these folds are used as training data and the remaining data are used as test data. Each of these different sets is tested, eventually producing *k* classifiers. The performance measure that is used in this method is the average of the calculated values. Due to the relatively high performance offered by cross-validation and to ensure that all selected categories are represented across all folds, many of the literature recommends its use, although it is not commonly used in most experiments because cross-validation requires more data and higher specification computers [140]. Figure 4.17 shows how the *k*-fold cross-validation method works.

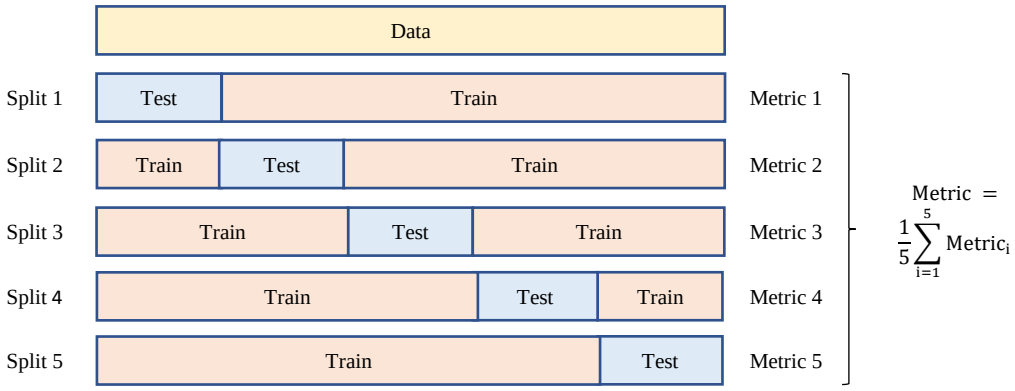

Figure 4.17. K-fold cross-validation method for training a ML model

It is possible to divide the training data according to the time periods [95, 141], where the oldest data is used for training and the newly added data is used for testing. This type is usually used when for actual classification purposes, where there is an actual need to classify the data and not just to evaluate the results, since the only data

available in a text operational classification classifier has been produced so far. This method also requires adding the date of creation of the documents as well so that the classifier can deal with them appropriately. Most important in all ways is to keep the training and test sets separate from each other, to avoid misleading good results.

In this thesis we will use the two most common types of training a ML model, the hold out method and the *k*-fold cross-validation method. In the first stage of results, the filter FS method, we will use the *k*-fold cross-validation method because this method is able to give reliable results despite the longer time it takes compared to the hold out method. There is no fixed rule for choosing the number of folds, but most researchers agreed on the use of the value of *k* between 5-10 [142, 143], for us we will use *k=10* in this thesis. A lower value of *k* results in a large similarity in the way it works, and thus the results, with the hold out method, so choosing a higher *k* makes more sense. The higher the *k* value, the smaller the difference between the size of the training data and the test data, and the lower this difference becomes, the smaller the bias and noise [144, 145]. In the second stage of results, the wrapper FS method, our use of the GA will largely ensure that we use all the available data, so the need to use the *k*-fold cross-validation method is less than in the first case, so we will use a holdout method with the GA. In a GA, with a large number of generations being used, it would be computationally very difficult to use the *k*-fold cross-validation method as it would require a very long time and high-spec hardware, despite all this, the results that we will get by using the hold out method may be better and require much less time.

4.10. Evaluation Metrics

In the literature, it is preferable to have many comparable performance measures to compare the performance of different TW schemes, classifiers, and algorithms. The main problem when evaluating TC methods is the lack of standard protocols for data collection, meaning that the data are usually different and some data may contain more random data or noise than others, and this generates large differences in the results of assessment methods. Even in the case of a common method of data collection, simply changing the size of the training and testing sets can lead to inconsistencies in model performance [146]. Performance metrics generally evaluate certain aspects of a rating task, with each metric focusing on a particular aspect, and thus not always providing identical information. Because the underlying mechanisms of different rating scales

may differ, understanding exactly what each of these scales represent and the type of information they are trying to convey is critical to comparability. In this thesis results will be compared using common performance measures, recall, precision, and f-measurement. These metrics are based on the confusion matrix, shown in Table 4.8

Recall (sometimes called sensitivity) represents the model's ability to find all relevant cases in the data set, as shown in (4.43).

$$Recall = \frac{TP}{TP + FN} \tag{4.43}$$

Precision represents the model's ability to identify only the relevant data points, as shown in (4.44).

$$Precision = \frac{TP}{TP + FP} \tag{4.44}$$

The f1-score of the system is defined as the weighted harmonic means of its precision and recall. It is defined as follows,

$$F1 - score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{4.45}$$

In the case of binary classification, we can use the performance measures as we mentioned in the above equations, but this evaluation metrics cannot be used literally in multi-label classification, Therefore, evaluation metrics should be used in the form of micro, macro and weighted. Micro-average score used to weight each condition, predicted equally. Macro-average score used when all classes need to be treated equally to evaluate the overall performance of the classifier with respect to the most common class labels. Weighted-average score in category disequilibrium state (a different number of documents in each category) [147].

For precision there are two common methods of use, the first type is micro-average precision which can be defined as the sum of the *TP* of all the groups divided by all the positive predictions. A positive prediction is the sum of *TP* and *FP*. It can be calculated mathematically by the following equation,

$$PrecisionMicroAvg = \frac{(TP_1 + TP_2 + \cdots + TP_n)}{(TP_1 + TP_2 + \cdots + TP_n + FP_1 + FP_2 + \cdots + FP_n)} \quad \textbf{(4.46)}$$

The second type, macro-average precision, can be calculated as the arithmetic mean of all precision for the different classes. It can be calculated mathematically by the following equation,

$$PrecisionMacroAvg = \frac{(Prec_1 + Prec_2 + \cdots + Prec_n)}{n} \quad \textbf{(4.47)}$$

For Recall there are also two common methods of recall, the first type is micro-average recall which can be defined as the sum of the *TP* of all the categories divided by the actual positives (not the expected positives). It can be calculated as,

$$RecallMicroAvg = \frac{(TP_1 + TP_2 + \cdots + TP_n)}{(TP_1 + TP_2 + \cdots + TP_n + FN_1 + FN_2 + \cdots + FN_n)} \quad \textbf{(4.48)}$$

The second type, the macro-average recall, can be calculated as the arithmetic mean of all the recall scores for the different classes. It can be calculated mathematically as,

$$RecallMacroAvg = \frac{(Recall_1 + Recall_2 + \cdots + Recall_n)}{n} \quad \textbf{(4.49)}$$

The most widely used evaluation metric in the literature remains the f1-score. F1 represents the harmonic mean of precision and recall as shown (4.45). Because f1-score considers only precision and recall, Intuitively, an f1-score is only high if both scales are high, maxes out when both scales point to 1, but is low if one of the two scales has a low score. When we want to calculate f1-score for multi-class data, the average parameter in the f1-score function needs to be one of these,

Macro-average f1-score, which calculates the f1-score for each category individually and then adds the values to sum the categories. This model performs well if the groups are equal but leads to bad results if the groups are unequal, and the model has performed poorly with minority groups. This type is calculated mathematically as follows,

$$F1MacroAvg = F1_{Class\ 1} + F1_{Class\ 2} + \cdots + F1_{Class\ n} \quad \textbf{(4.50)}$$

Micro-average f1-score, this type calculates the value of f1-score directly as this type does not favor one category over the other. This type is calculated mathematically as,

$$F1MicroAvg = F1_{Class\ 1+Class\ 2+\cdots+Class\ n} \quad \textbf{(4.51)}$$

Weighted-average f1-score, this type calculates the f1-score for each category separately, but when combined, it multiplies the f1-score of each category by the weighted value of this category, which depends on the number of true labels of each class, so This type is preferred by the majority category. This type is calculated mathematically as,

$$F1weightedAvg = (F1_{Class\ 1} \cdot W_1) + (F1_{Class\ 2} \cdot W_2) + \\ \ldots + (F1_{Class\ n} \cdot W_n) \quad \textbf{(4.52)}$$

Which type to use depends on the results we want to achieve. Macro-average is usually used in statements where there is concern about class imbalance. We can summarize the previous details by saying that macro-average deals with every problem equally, so it is considered the arithmetic average of all categories, micro-average cares more about documents, that is, it deals with every document equally, weighted-average gives importance to each category according to the number of documents it has belong to each category. For real applications and problems that contain many documents in many unequal categories, each method will show different results than the other. In this thesis, we will use an equal dataset, meaning that all classes contain an equal number of documents so the values for macro-average, micro-average and weighted average will be very similar. Since we must choose one of the methods to use in our results, we will use the macro-average in all the results that we will get in this thesis.

5. HYBRID FS-GA APPROACH

To get good results from the classification of a difficult language such as Arabic, the classification process must go through many steps, as each step will contribute to the success of the classification process and obtaining the results we desire. This chapter will show the steps that our proposed approach will follow to obtain the results.

5.1. Input Data

Selecting the appropriate data is one of the most important stages of building any TC model, as the data will directly affect the model performance. Many Arabic datasets have been created for use in many applications and studies. Due to the lack of available high-quality Arabic datasets and the desire to use a carefully selected set of data collection and processing techniques, this thesis presents a new Arabic dataset created using web scraping techniques. The web has become the largest source of data in the world, especially with the growth of the fields of big data analytics, AI, and ML, which called on researchers to develop and exploit increasingly web scraping methods. Web scraping is a form of data mining for extracting unstructured data from websites and turning it into structured data that can be stored and analyzed [148], as shown in Figure 5.1.

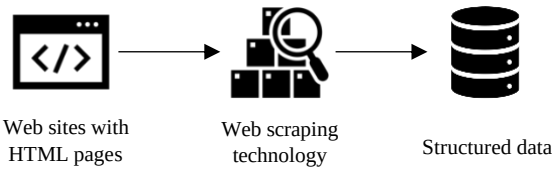

Figure 5.1. Structure of the web scraping

Document object model (DOM) analysis technology is a popular web scraping technique. This technique creates a parsing tree for the parsed pages that can be used to extract data from HTML, which is useful for web scraping. Figure 5.2 shows one of the news websites that web scraping technology can be used on (Al-Jazeera news website - Political News Section). The page URL, shown at the top of the figure, is used to determine the site that will be scraped, as the news title, its details, the date of the news publication, and the news category will be taken.

Figure 5.2. Information available on the news site page with html source

After we scraped sites, a folder was created for each scraped site, each folder containing subfolders titled categories or labels. Each subfolder contains a file in which news information is saved such as the title of the news, its details, publication date, and its category. After completing the preparation of all the folders, all the data is collected in one large file that we called ultimate Arabic news dataset (UAND) [149, 150]. Each news category was checked separately before it was added to the final data set to ensure that the UAND dataset contained the highest quality data. The main UAND dataset consist of more than 200,000 text documents divided into nine news categories: sports, politics, art, culture, economy, religion, medicine, technology, and society. In this thesis, due to time constraints and hardware specifications that were used, four versions of the data were used, each version contains a subset of documents consisting of 2700 text documents where each version contains different pre-processing steps to select the appropriate dataset to be used in the following experiments. Table 5.1 shows additional information about the versions.

Table 5.1. UAND versions

Data No.	Dataset name	Data Cleaning	Stop word filtering	Data Stemming	Number of documents	Number of all tokens
1	Original	NO	NO	NO	2700	100543
2	Original without Stop	YES	YES	NO	2700	82876
3	Original with Stem	YES	NO	YES	2700	48926
4	Original pre pros.	YES	YES	YES	2700	42462

The first version (original) contains the raw data without any pre-processing, so we notice that the number of tokens is very high. The preprocessing techniques mentioned in Section 4.4, which is data cleaning, stop word removal, and text stemming technique, have been applied to the other three versions. In the second version (original without stop) the data was cleaned and stop words was removed, so we notice that the number of symbols is greatly reduced. In the third version (original with stem) the data was cleaned, and text stemming technique was used to remove all additions and suffixes. In the fourth version (original pre pros), all preprocessing techniques such as data cleaning, stop word removal and text stemming technique were applied, so we note that the number of tokens is the lowest among all versions. UAND in all its versions has been made publicly available to researchers and those interested in Arabic TC. All UAND data is ready for direct classification as it has been collected with high accuracy as the results in this thesis indicate [149, 150].

5.2. Hybrid FS-GA Approach

GA is a good FS technique, but when the search space is the feature dimension space, GA suffers from major difficult challenges of time consumption, setting the appropriate parameters, and selecting the features most of which will be noisy and irrelevant features, which negatively affects the classification performance [151]. GA suffers from some problems, including that it is impractical when dealing with high-dimensional texts as it takes a long time, and randomization in the initial population, which may affect the result. To avoid these problems and ensure good results in the FS and TC process in general, we suggest using the TW schemes in a two-stage process, the filter stage, and the GA stage.

5.2.1. Filter FS methods

Filter FS is the most widely used method in research, as the best terms found by TW schemes are used directly with the classifiers to find results. At this stage, we will use 5 unsupervised TW schemes, 16 supervised TW schemes, and 4 classifiers mentioned in the previous chapter. In this part of the proposed system, the appropriate data version and the appropriate number of features will be selected for use in the following experiments, as the number of features should not be few, so that some good features are lost, and not be large, as it includes many random features. After choosing the

appropriate number of features, the results are obtained in detail to select the 5 best TW schemes and the best classifier to be used with GA. This stage is implemented for three purposes, first, to reduce large text dimensions that can cause major problems for the GA, second, to reduce the impact of random features that reduce the performance, and third, to speed up the process of selecting the best features with higher accuracy.

5.2.2. GA based wrapper method

The best features that were found by the best TW schemes are selected, in other words, the features are arranged in a descending manner where the highest, and therefore the best, features are chosen, the least important features are ignored and thus we ensure that the dimensions of the features that the GA deals with are reduced and to ensure that the features we use are high quality features and from which we are able to obtain a subset of the features with high accuracy. The features found using the best TW schemes are used to categorize the documents, which will ensure that we use features that are few in number and that are of high quality. To understand how the GA works in our proposed system, we will use an example with only 10 features, which are very few and do not happen in practical applications, but the point of this example is just to show how the GA deals with features. The words we will be using are *ball*, *MP*, *Painting*, *God*, *barrel*, *weather*, *civilization*, *invention*, *doctor*, and *foot*. For the parameters, the number of generations will be 4 and the population size will be 6.

GA starts by creating the first generation, which is a population matrix whose size is equal to the number of populations. In the parameters, the number of features that will be used is determined from the total number of available features. In our case for this example, we will reduce the number of features from 10 features to 5, meaning that 50% of the number of available features will be used. Each chromosome must be completely different from the other, that is, no two individuals can be completely identical. Table 5.2 shows an example population matrix for our example.

Table 5.2. Population matrix for first generation

Chromosome	Feature									
	Foot	*Ball*	*MP*	*Painting*	*God*	*Barrel*	*Weather*	*Civilization*	*Invention*	*Doctor*
1	0	1	1	0	1	0	1	1	0	0
2	1	0	1	1	0	1	0	0	1	0
3	0	0	0	0	1	1	1	1	0	1
4	0	0	1	1	0	0	0	1	1	1
5	1	1	0	0	1	0	1	0	1	0
6	1	1	0	1	0	1	0	0	0	1

GA start searching for potential parents for the next generation by classifying the text using features selected from the population matrix and one of the classifiers mentioned in section 4.7.2. The f1-score for each individual in Table 5.2 will be as follows,

$$F1_{Terms_1} = F1_{[Ball,MP,God,Weather,Civilization]} = 0.750$$
$$F1_{Terms_2} = F1_{[Foot,MP,Painting,Barrel,Inventation]} = 0.767$$
$$F1_{Terms_3} = F1_{[God,Barrel,Weather,Civilization,Doctor]} = 0.739$$
$$F1_{Terms_4} = F1_{[MP,Painting,Civilization,Inventation,Doctor]} = 0.776$$
$$F1_{Terms_5} = F1_{[Foot,Ball,God,Weather,Inventation]} = 0.857$$
$$F1_{Terms_6} = F1_{[Foot,Ball,Painting,Barrel,Doctor]} = 0.662$$

In the selection stage, GA select the individuals according to the selection probability of each individual. GA select individuals by one of the selection methods, in this thesis, we used most used selection method which is roulette wheel selection method.

Table 5.3. Roulette wheel selection method

Chromosome	Fitness (F1 score)	Probability of choice	Selected individuals
1	0.750	16.5%	
2	0.767	16.9%	◀
3	0.739	16.2%	
4	0.776	17.1%	
5	0.857	18.8%	◀
6	0.662	14.5%	

Using roulette wheel selection method, second and fifth chromosomes are chosen. Depending on the individuals (parents) who were selected from the selection stage, we will produce the next generation of offspring as follows in the crossover stage,

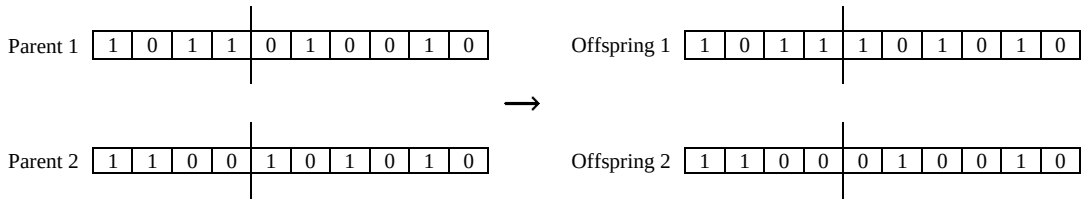

Figure 5.3. First single-point crossover attempt

Figure 5.3 shows that the number of genes (features) selected in the new chromosomes (offspring) may be greater than the number of genes selected in the parents. After several generations, the number of genes (features) may deviate significantly from the specified number. To reduce this deviation, swap the “0” bit to “1” to increase the

number of features to the specified number of features, or swap the “1” bit to “0” to decrease the number of features to the specified number of features.

This method may be good for problems with few features, but in this message a large number of features will be used so this traditional method may cause the loss of a large number of good features and it is a process similar to the random mutation process. To get rid of this problem, a good method will be used to ensure that the number of features does not deviate significantly from the specified number of features, and at the same time give some freedom to the genetic algorithm to perform its normal work and not be restricted by strict laws. The genetic algorithm in our proposed system will be given a deviation range of 1%-3% from the given number of features. If the number of features increases, or decreases, from the specified field, another intersection point will be chosen to ensure that the number of features does not exceed the specified number. Figure 5.4 shows the second attempt to find the new individuals after the number of features exceeds the specified number of features, as shown in Figure 5.1.

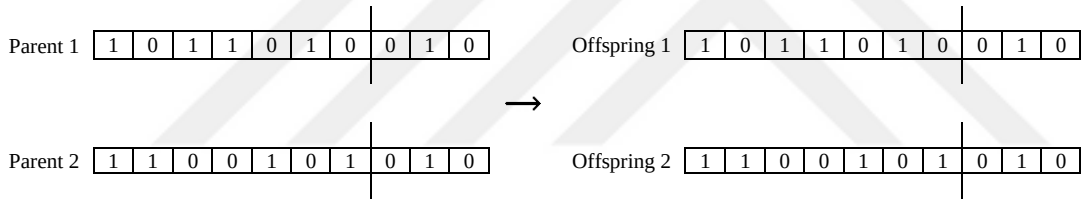

Figure 5.4. Second single-point crossover attempt

We note that the number of selected features in each chromosome became within the specified range after changing the crossover point from the fourth to the seventh position. After the completion of the crossover process, the resulting individuals are transferred to the mutation stage.

GA generate two random values to determine the probability of a mutation in the generation, the first value being the mutation probability which is very small, usually 0.005 to 0.1, and a second value between 0 and 1, if the mutation probability is greater than the random number, then apply the mutation steps mentioned in section 4.8.2. If the random number of the first child is less than the probability of mutation, the mutation will be applied to the first child as shown in Figure 5.5.

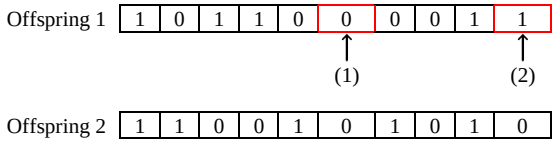

Figure 5.5. Second single-point crossover attempt

If the number of genes exceeds the maximum or minimum number of genes after the mutation process, another gene is selected and converted from "0" bit to "1" bit or "1" bit to "0" as needed. After completing this step, we will have two new individuals produced, so an empty population matrix is created, the two new individuals are added to it, as shown in Table 5.4.

Table 5.4. New population matrix for second generation

Chromosome	Feature									
	Foot	*Ball*	*MP*	*Painting*	*God*	*Barrel*	*Weather*	*Civilization*	*Invention*	*Doctor*
1	1	0	1	1	0	0	0	0	1	1
2	1	1	0	0	1	0	1	0	1	0
3										
4										
5										
6										

The previous steps are repeated until the new population matrix is full. When the new matrix is filled and the current generation is finished, GA check the stopping conditions. If one of the stopping conditions is met, GA end the algorithm and get the best result obtained, and if none of the stopping conditions are met, GA repeats the steps until the stopping conditions are met. Figure 5.6 shows hybrid FS-GA approach that we will use in this thesis to obtain good results.

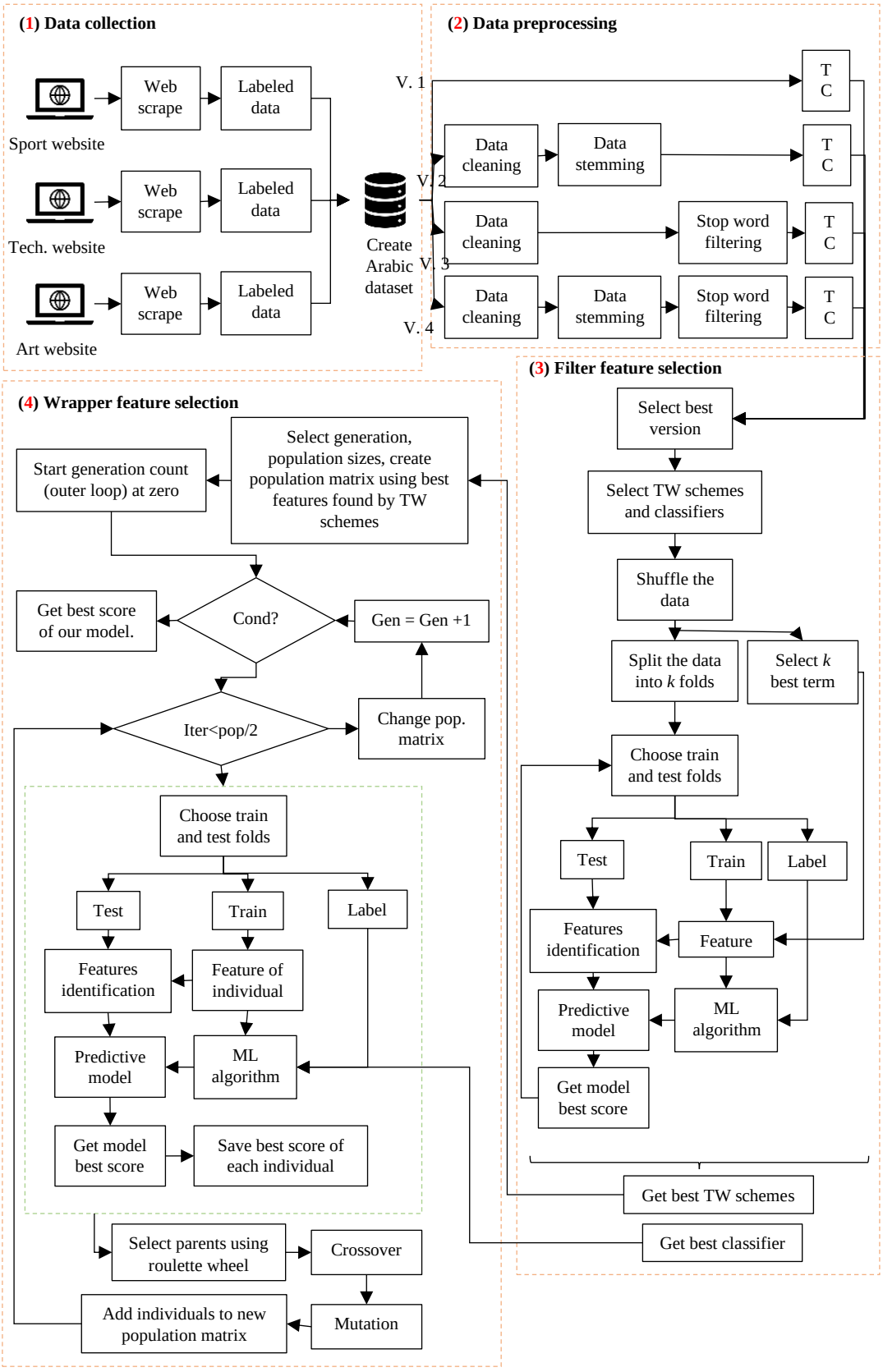

Figure 5.6. Hybrid FS-GA approach

6. RESULT AND DISCUSSION

This chapter discusses the results obtained using the filter FS method and hybrid FS-GA approach. 21 TW schemes and four classifiers are used to select the best features in the filter FS method, and these techniques were combined with GA to construct a hybrid FS-GA approach.

6.1. Experiment Setup

This section describes the environment in which the experiments is performed, in terms of the hardware and software environment in the first section, and the classifiers used to evaluate the performance in the second section.

6.1.1. Hardware and software

One of the most important factors in finding optimal results using algorithms is the quality of the device that is being worked through, as the high specifications device provides the ability to deal with a larger amount of data and features and provides the ability to obtain more quality results in less time. This thesis is performed on a 4GB memory device with an AMD E1-2100 APU with Radeon (TM) HD Graphics 1.00 GHz. Python 3.8.6 on windows 10 64-bit was used to get the numerical results.

6.1.2. Combination experiments

Four supervised ML techniques will be used for TC, two parametric ML algorithms, LR and Bernoulli NB, and two non-parametric ML algorithms, Linear SVC and RF. Default parameters were used for all classifiers, as it showed promising results, so we preferred to continue using them instead of using new parameters. The first classifier, Linear SVC, is a very similar classifier to SVM with the parameter kernel= “linear”, but in this thesis, linear SVC was used because of its ability to handle large data numbers with greater flexibility and stability [114]. For the second classifier, Bernoulli NB classifier was chosen because of its high ability to handle binary features. LR was chosen because its ability to handle both dense and sparse input. RF was chosen because it uses averaging to improve the predictive accuracy and control over-fitting. First, each term was scored using the TW schemes that we mentioned in section 4.5, and then we selected the highest *n* features to use with our chosen algorithms. We chose a large number of TW schemes and algorithms due to the fact that each TW

scheme and each classifier can affect classification results directly. All filter FS experiments were performed using 10-fold cross-validation, while wrapper FS experiments were performed using train-test validation. Using 10-fold cross-validation helps ensure more stable results with filter FS, as there are no multiple attempts to find the best results as in wrapper FS methods, and experiments are often repeatable. In contrast, a train-test split in the experiments with wrapper FS are used, where multiple iterations of the GA ensure that the most data are used, and the train-test split helps get results faster. These many techniques included in the proposed approach helps produce many results with high quality and good stability even using a very small number of features. All experiments were repeated five times to prevent any random results that may occur and to confirm the accuracy of the results obtained. For the genetic algorithm that was used with the hybrid FS-GA approach, Table 6.1 shows the parameters that were used and showed very good results.

Table 6.1. The parameter values for GA

GA Parameter	Value
Generations	70
Population size	50
Elite size	1
Selection method	Roulette wheel selection
Crossover type	Single-Point crossover
Crossover Probability	0.9
Mutation Type	One bit Mutation
Mutation Probability	0.1
Number of documents	2700 (9 categories)

6.2. Dataset Results

It was mentioned in Section 5.1 that four different versions of the UAND dataset will be tested to choose the best version for use in upcoming experiments. Table 6.2 shows the full results of the UAND versions using the best 1000 features found using IG, MI and CHI2, three of the most widely used TW schemes, with Bernoulli NB, LR, linear SVC, and RF.

Table 6.2. F1-score results with UAND versions

Classifier	IG				MI				CHI2			
	V. 1	***V. 2***	***V. 3***	***V. 4***	***V. 1***	***V. 2***	***V. 3***	***V. 4***	***V. 1***	***V. 2***	***V. 3***	***V. 4***
Linear SVC	**0.879**	0.808	0.800	0.825	0.802	0.808	0.821	**0.826**	0.860	0.830	**0.867**	0.832
LR	0.641	**0.776**	0.692	0.755	0.717	0.773	0.693	**0.803**	0.770	0.750	0.767	**0.781**
Bernoulli NB	0.788	0.825	0.841	**0.844**	0.835	**0.860**	0.821	0.818	0.818	0.797	0.797	**0.839**
RF	0.842	0.790	0.860	**0.872**	**0.855**	0.819	0.807	0.832	**0.865**	0.832	0.788	0.834

Linear SVC showed that it was the classifier with the best f1-score results, as reached 0.879 using the best features found by IG with the first version and reached 0.867 using the best features found by CHI2 with the third version. With a very small difference, RF showed very good f1-score results with an 0.872 using the best features found by IG with the fourth version. Bernoulli NB showed very similar f1-score results for the two previous classifiers as well, reaching its peak performance of 0.860 using the best features found by MI with the second version. LR is the only classifier for which we can say that its performance was lower than other classifiers, as it reached its peak performance at 0.803 using the features found by MI with the fourth version. The three TW schemes we used showed good f1-score results, IG peaked at 0.879 with linear SVC and first data version, while the highest value for MI was 0.860 using Bernoulli NB and second data version, and the highest value for CHI2 was 0.867 using Bernoulli NB and third data version.

In the main part of this section of the experiments, which is to compare the performance of the UAND versions, the fourth version showed very good f1-score results, as the version was the best with LR and RF with 0.803 and 0.872, respectively, and the second-best version when used with LR and Bernoulli NB with 0.781 and 0.844, respectively. The first version also showed good f1-score results, as it was the best version when used with linear SVC with a f1-score of 0.879, and the second-best version when used with RF with a f1-score of 0.865. While the second version was the best version when used with Bernoulli NB with a f1-score of 0.860. The third version could not be the best version with any of the four classifiers that were used and was satisfied with being the second-best version when used with linear SVC with a f1-score of 0.867.

The objective of comparing the f1-score results of different UAND versions is to evaluate the data we collected, and to select the appropriate version for use in the upcoming experiments because using all four versions will create too many combinations to illustrate or discuss. Given the results we obtained, the fourth data version, which includes all the preprocessing methods, will be used in upcoming experiments.

6.3. Number Of Features

The purpose of this experiment is to determine the optimal number of features that produce the best results, so a wide range of the number of features ranging from 100 to 10,000 features was tested. Table 6.3 shows the performance all TW schemes, described in Section 4.5, with different number of features when the linear SVC is the classifier.

Table 6.3. F1-score of different number of features found by TW schemes with the linear SVC

Type	TW scheme	100	500	1000	2000	4000	6000	8000	10000
Un-Supervised	Random	0.045	0.123	0.216	0.293	0.425	0.496	0.516	**0.574**
	DF	0.653	0.801	0.842	0.846	0.868	0.875	0.866	**0.883**
	CF	0.687	0.832	0.864	0.857	0.861	0.863	**0.875**	0.865
	CF-IDF	0.699	0.820	0.857	0.861	0.863	0.867	**0.869**	0.867
	TF-IDF	0.733	0.829	0.865	0.847	**0.877**	0.856	0.861	0.861
Supervised	IG	0.739	0.842	0.875	0.869	**0.881**	0.868	0.869	0.876
	MI	0.728	0.831	0.864	0.868	**0.891**	0.873	0.868	0.864
	CHI2	0.716	0.824	0.846	0.868	0.882	0.876	0.874	**0.885**
	OR	0.403	0.580	0.625	0.692	0.776	0.778	0.786	**0.828**
	GR	0.710	0.840	0.851	**0.879**	0.870	0.872	0.874	0.870
	CPD	0.406	0.560	0.648	0.678	0.760	0.775	0.813	**0.832**
	ICF	0.262	0.676	0.805	0.837	0.863	0.869	0.882	**0.884**
	TF-IDF-ICF	0.611	0.815	0.839	0.872	0.866	**0.884**	0.873	0.862
	IGM	0.317	0.658	0.741	0.775	0.813	0.842	0.851	**0.862**
	RF	0.402	0.553	0.605	0.639	0.720	0.709	0.755	**0.762**
	IDFEC	0.724	0.845	0.874	0.865	0.871	0.856	**0.883**	0.869
	GI	0.687	0.823	0.874	0.869	0.880	0.851	**0.893**	0.873
	GT	0.618	0.833	0.862	**0.886**	0.879	0.880	0.876	0.865
	GSS	0.764	0.836	0.861	0.869	0.859	0.880	**0.882**	0.860
	NGL	0.697	0.834	0.864	0.870	0.875	**0.888**	0.879	0.868
	DFS	0.492	0.646	0.699	0.756	0.764	0.804	0.815	**0.821**

The f1-score in Table 6.3 shows that most of the TW schemes reached their peak performance using a large number of features ranging from 8000-10000 features, but there are some TW schemes that showed very good f1-score results using a smaller number of features. TF-IDF was the best unsupervised TW scheme used as it reached its peak performance of 0.877 using the best 4000 features and 0.865 using the best 1000 features. For supervised TW schemes, IG and MI reaching their peak f1-score of 0.881 and 0.891 respectively using the top 4000 features, and GR and GT reaching their peak performance of 0.879 and 0.886 using the top 2000 features. Although the use of a small number of features may cause the loss of some good features, some schemes were able to achieve excellent f1-score results using a few features such as IG whose performance reached 0.875 using the top 1000 features and IDFEC whose performance reached 0.874 and 0.845 using the best 1000 and 500 features, respectively. Some other schemes showing impressive upward performance are CHI2, GSS and NGL.

Results show that the f1-score began to rise when the number of features exceeded 1000 features, which shows that many TW schemes were able to extract many good features in the top 1000 features. The performance starts to decline clearly when the number of features is less than 1000 features. This thesis aim to achieve the best possible performance using as few features as possible, so we will use the top 1000 features found using all TW schemes as a starting point and try to find fewer features with higher performance.

6.4. Filter FS Method Results

After selecting the appropriate number of features, these features are used to select a smaller number of features ranging from 1%-100% of the original number of features. Table 6.4 shows the recall, precision, and f1-score and Table 6.5 shows the time required to obtain the results when used with all four classifiers.

Table 6.4. Precision, recall, and f1-score values for all TW schemes when used with the four classifiers at different sizes of features

TW	Subset size	Linear SVC			LR			Bernoulli NB			RF		
		Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
Random	1%	0.111	0.010	0.019	0.111	0.011	0.020	0.111	0.010	0.018	0.111	0.011	0.020
	2.5%	0.111	0.011	0.021	0.112	0.021	0.019	0.111	0.010	0.018	0.112	0.015	0.019
	5%	0.118	0.092	0.032	0.115	0.103	0.029	0.112	0.056	0.024	0.112	0.045	0.024
	10%	0.115	0.086	0.027	0.114	0.073	0.025	0.116	0.151	0.032	0.118	0.140	0.035
	30%	0.150	0.337	0.095	0.160	0.269	0.108	0.161	0.317	0.110	0.159	0.288	0.108
	50%	0.174	0.309	0.132	0.181	0.364	0.140	0.168	0.318	0.125	0.181	0.366	0.366
	70%	0.226	0.504	0.219	0.214	0.453	0.453	0.197	0.476	0.183	0.207	0.489	0.195
	90%	0.234	0.469	0.225	0.226	0.491	0.228	0.239	0.481	0.238	0.225	0.469	0.214
	100%	0.246	0.539	0.243	0.262	0.531	0.262	**0.246**	0.519	0.236	0.246	0.496	0.244
DF	1%	0.289	0.294	0.262	0.307	0.330	0.291	0.290	0.293	0.255	0.274	0.288	0.256
	2.5%	0.426	0.443	0.417	0.426	0.444	0.423	0.443	0.459	0.436	0.450	0.462	0.443
	5%	0.558	0.562	0.543	0.535	0.551	0.531	0.556	0.580	0.553	0.558	0.561	0.552
	10%	0.663	0.672	0.659	0.661	0.667	0.659	0.675	0.690	0.671	0.681	0.688	0.676
	30%	0.779	0.781	0.776	0.758	0.759	0.752	0.778	0.797	0.777	0.772	0.775	0.767
	50%	0.815	0.820	0.812	0.785	0.790	0.776	0.811	0.829	0.811	0.822	0.828	0.822
	70%	0.833	0.834	0.828	0.811	0.819	0.808	0.808	0.842	0.813	0.817	0.823	0.823
	90%	0.824	0.830	0.821	0.813	0.818	0.812	0.827	0.857	0.831	0.809	0.813	0.817
	100%	0.844	0.844	**0.839**	0.814	0.816	0.810	0.832	0.851	0.830	0.836	0.843	0.834
CF	1%	0.296	0.281	0.255	0.310	0.322	0.265	0.303	0.351	0.280	0.288	0.313	0.265
	2.5%	0.420	0.454	0.410	0.419	0.444	0.414	0.421	0.443	0.409	0.389	0.420	0.383
	5%	0.619	0.652	0.615	0.612	0.630	0.609	0.619	0.641	0.617	0.603	0.613	0.601
	10%	0.697	0.712	0.696	0.695	0.694	0.688	0.697	0.714	0.694	0.697	0.712	0.692
	30%	0.811	0.815	0.809	0.779	0.783	0.774	0.814	0.828	0.812	0.821	0.827	0.818
	50%	0.831	0.836	0.830	0.792	0.810	0.789	0.830	0.844	0.829	0.827	0.834	0.827
	70%	0.838	0.847	0.837	0.812	0.829	0.809	0.831	0.842	0.828	0.838	0.846	0.836
	90%	0.850	0.857	0.847	0.792	0.810	0.784	0.849	0.866	0.851	0.826	0.827	0.822
	100%	0.863	0.875	**0.863**	0.820	0.837	0.816	0.853	0.865	0.852	0.852	0.849	0.848
CF-IDF	1%	0.307	0.261	0.245	0.309	0.281	0.250	0.318	0.394	0.271	0.291	0.306	0.245
	2.5%	0.428	0.489	0.418	0.415	0.446	0.405	0.412	0.474	0.408	0.434	0.486	0.424
	5%	0.598	0.619	0.591	0.592	0.607	0.583	0.607	0.634	0.607	0.584	0.594	0.579
	10%	0.704	0.714	0.704	0.695	0.702	0.690	0.698	0.721	0.698	0.705	0.710	0.701
	30%	0.805	0.805	0.801	0.799	0.810	0.797	0.796	0.818	0.796	0.806	0.814	0.804
	50%	0.828	0.834	0.827	0.818	0.826	0.814	0.825	0.845	0.825	0.825	0.831	0.825
	70%	0.838	0.849	0.840	0.817	0.828	0.813	0.846	0.860	0.848	0.834	0.840	0.833
	90%	0.853	0.853	0.849	0.813	0.832	0.810	0.845	0.859	0.845	0.838	0.842	0.836
	100%	0.864	0.871	**0.863**	0.808	0.828	0.801	0.851	0.861	0.849	0.839	0.841	0.836

TF-IDF	1%	0.350	0.420	0.335	0.373	0.414	0.359	0.362	0.380	0.335	0.372	0.436	0.356
	2.5%	0.527	0.552	0.525	0.561	0.591	0.555	0.539	0.566	0.532	0.570	0.594	0.565
	5%	0.662	0.667	0.658	0.648	0.660	0.647	0.662	0.689	0.662	0.663	0.669	0.659
	10%	0.744	0.753	0.743	0.729	0.741	0.731	0.713	0.754	0.717	0.736	0.744	0.736
	30%	0.817	0.828	0.828	0.812	0.824	0.811	0.806	0.836	0.810	0.799	0.813	0.801
	50%	0.840	0.846	0.839	0.796	0.816	0.796	0.837	0.857	0.838	0.833	0.840	0.833
	70%	0.858	0.864	0.856	0.810	0.826	0.807	0.858	0.872	**0.872**	0.835	0.842	0.833
	90%	0.859	0.868	0.859	0.843	0.852	0.836	0.865	0.883	0.868	0.846	0.849	0.843
	100%	0.866	0.876	0.867	0.827	0.851	0.827	0.860	0.876	0.863	0.847	0.851	0.846
IG	1%	0.352	0.355	0.298	0.363	0.330	0.302	0.374	0.429	0.345	0.360	0.422	0.335
	2.5%	0.530	0.534	0.495	0.551	0.538	0.508	0.549	0.630	0.532	0.542	0.630	0.531
	5%	0.671	0.732	0.663	0.640	0.700	0.637	0.680	0.738	0.684	0.657	0.706	0.653
	10%	0.729	0.751	0.725	0.732	0.752	0.726	0.753	0.792	0.759	0.758	0.772	0.757
	30%	0.800	0.803	0.797	0.792	0.789	0.785	0.818	0.841	0.821	0.799	0.817	0.801
	50%	0.844	0.844	0.841	0.820	0.827	0.818	0.820	0.835	0.819	0.826	0.832	0.823
	70%	0.855	0.853	0.852	0.837	0.840	0.834	0.851	0.863	0.850	0.849	0.853	0.848
	90%	0.864	0.866	0.860	0.842	0.846	0.837	0.840	0.857	0.839	0.857	0.859	0.854
	100%	0.875	0.872	**0.871**	0.844	0.849	0.841	0.868	0.878	0.869	0.855	0.852	0.850
MI	1%	0.347	0.359	0.307	0.329	0.292	0.261	0.358	0.403	0.320	0.342	0.402	0.309
	2.5%	0.522	0.533	0.490	0.520	0.527	0.483	0.531	0.560	0.511	0.540	0.605	0.525
	5%	0.662	0.679	0.635	0.662	0.703	0.639	0.669	0.736	0.674	0.655	0.688	0.644
	10%	0.737	0.756	0.736	0.729	0.747	0.721	0.766	0.800	0.768	0.755	0.778	0.755
	30%	0.803	0.800	0.797	0.792	0.798	0.790	0.814	0.843	0.817	0.792	0.794	0.788
	50%	0.843	0.841	0.836	0.816	0.817	0.812	0.860	0.870	0.858	0.832	0.837	0.830
	70%	0.842	0.845	0.840	0.828	0.836	0.825	0.851	0.859	0.847	0.849	0.847	0.842
	90%	0.850	0.851	0.846	0.833	0.840	0.832	0.846	0.868	0.847	0.864	0.869	**0.862**
	100%	0.859	0.858	0.855	0.842	0.847	0.840	0.866	0.869	0.860	0.853	0.856	0.852
CHI2	1%	0.357	0.281	0.278	0.340	0.284	0.278	0.348	0.363	0.310	0.359	0.348	0.311
	2.5%	0.445	0.457	0.402	0.463	0.491	0.411	0.460	0.483	0.422	0.460	0.505	0.428
	5%	0.626	0.716	0.623	0.643	0.749	0.646	0.610	0.713	0.616	0.623	0.723	0.620
	10%	0.698	0.743	0.694	0.710	0.749	0.701	0.688	0.753	0.696	0.720	0.765	0.725
	30%	0.786	0.787	0.780	0.764	0.776	0.762	0.800	0.823	0.803	0.777	0.781	0.774
	50%	0.825	0.824	0.820	0.812	0.816	0.808	0.800	0.823	0.801	0.817	0.820	0.816
	70%	0.845	0.852	0.844	0.830	0.835	0.828	0.848	0.854	0.845	0.828	0.830	0.825
	90%	0.847	0.846	0.844	0.837	0.841	0.834	0.845	0.862	0.847	0.842	0.842	0.838
	100%	0.860	0.865	**0.859**	0.838	0.845	0.832	0.850	0.864	0.849	0.849	0.851	0.846
OR	1%	0.236	0.455	0.205	0.245	0.454	0.214	0.228	0.435	0.195	0.246	0.456	0.212
	2.5%	0.287	0.568	0.265	0.282	0.565	0.262	0.298	0.522	0.276	0.282	0.568	0.262
	5%	0.346	0.561	0.341	0.337	0.558	0.335	0.349	0.553	0.347	0.337	0.559	0.330
	10%	0.410	0.618	0.399	0.398	0.642	0.390	0.396	0.593	0.396	0.399	0.619	0.393
	30%	0.525	0.865	0.536	0.530	0.868	0.550	0.510	0.812	0.528	0.531	0.837	0.543
	50%	0.594	0.869	0.585	0.563	0.843	0.577	0.524	0.780	0.546	0.559	0.848	0.577
	70%	0.579	0.803	0.588	0.580	0.818	0.597	0.575	0.793	0.606	0.587	0.823	0.606
	90%	0.604	0.798	0.607	0.594	0.807	0.605	0.578	0.804	0.611	0.600	0.808	0.617
	100%	0.619	0.784	0.616	0.614	0.806	0.615	0.581	0.813	0.617	0.618	0.797	**0.627**
GR	1%	0.320	0.336	0.272	0.329	0.313	0.266	0.337	0.399	0.317	0.344	0.453	0.316
	2.5%	0.506	0.475	0.455	0.489	0.493	0.442	0.497	0.569	0.483	0.498	0.556	0.479
	5%	0.624	0.647	0.611	0.603	0.657	0.585	0.624	0.714	0.638	0.615	0.669	0.604
	10%	0.706	0.732	0.697	0.720	0.762	0.718	0.722	0.789	0.730	0.701	0.728	0.699
	30%	0.800	0.806	0.797	0.780	0.783	0.770	0.795	0.818	0.799	0.797	0.800	0.795
	50%	0.836	0.836	0.832	0.817	0.816	0.810	0.818	0.843	0.822	0.824	0.829	0.822
	70%	0.852	0.851	0.849	0.836	0.838	0.833	0.823	0.845	0.826	0.846	0.848	0.844
	90%	0.863	0.862	**0.860**	0.821	0.817	0.813	0.838	0.858	0.840	0.844	0.848	0.844
	100%	0.857	0.854	0.852	0.852	0.853	0.849	0.859	0.870	0.859	0.844	0.842	0.840
CPD	1%	0.202	0.452	0.169	0.201	0.451	0.172	0.219	0.456	0.193	0.194	0.452	0.161
	2.5%	0.275	0.491	0.260	0.269	0.563	0.259	0.271	0.518	0.261	0.275	0.523	0.263
	5%	0.341	0.565	0.340	0.356	0.561	0.351	0.344	0.568	0.339	0.351	0.568	0.352
	10%	0.410	0.629	0.398	0.407	0.607	0.394	0.405	0.659	0.401	0.403	0.587	0.395
	30%	0.510	0.875	0.525	0.505	0.874	0.521	0.492	0.814	0.518	0.512	0.842	0.526
	50%	0.563	0.861	0.574	0.580	0.857	0.596	0.538	0.817	0.574	0.575	0.861	0.606
	70%	0.571	0.827	0.589	0.584	0.835	0.601	0.548	0.801	0.582	0.560	0.836	0.582
	90%	0.586	0.800	0.601	0.590	0.831	0.608	0.568	0.802	0.603	0.574	0.806	0.594
	100%	0.620	0.812	0.623	0.635	0.835	**0.654**	0.583	0.822	0.624	0.600	0.796	0.624
ICF	1%	0.135	0.371	0.072	0.138	0.302	0.071	0.134	0.323	0.063	0.138	0.369	0.073
	2.5%	0.174	0.437	0.129	0.166	0.460	0.121	0.177	0.503	0.134	0.169	0.547	0.124
	5%	0.231	0.572	0.206	0.238	0.504	0.215	0.229	0.505	0.203	0.212	0.591	0.180
	10%	0.288	0.682	0.276	0.286	0.659	0.267	0.279	0.683	0.266	0.295	0.728	0.288
	30%	0.547	0.821	0.579	0.527	0.797	0.562	0.534	0.819	0.580	0.518	0.809	0.556
	50%	0.669	0.793	0.685	0.674	0.790	0.689	0.633	0.814	0.666	0.633	0.769	0.657
	70%	0.734	0.798	0.741	0.747	0.807	**0.807**	0.698	0.823	0.725	0.695	0.791	0.713

	90%	0.759	0.793	0.764	0.785	0.822	0.787	0.754	0.854	0.775	0.721	0.784	0.729
	100%	0.801	0.825	0.802	0.789	0.818	0.788	0.773	0.857	0.793	0.750	0.786	0.752
TF-IDF-ICF	1%	0.300	0.505	0.265	0.305	0.504	0.270	0.307	0.493	0.268	0.310	0.481	0.270
	2.5%	0.399	0.637	0.388	0.396	0.630	0.389	0.394	0.631	0.377	0.387	0.643	0.377
	5%	0.491	0.756	0.521	0.505	0.754	0.534	0.474	0.730	0.505	0.495	0.725	0.520
	10%	0.588	0.723	0.603	0.600	0.732	0.615	0.574	0.728	0.602	0.593	0.725	0.618
	30%	0.789	0.791	0.784	0.746	0.746	0.740	0.749	0.787	0.749	0.750	0.758	0.748
	50%	0.825	0.824	0.821	0.797	0.799	0.791	0.810	0.834	0.809	0.774	0.783	0.774
	70%	0.844	0.848	0.843	0.824	0.827	0.818	0.833	0.850	0.836	0.812	0.818	0.811
	90%	0.859	0.864	**0.858**	0.818	0.829	0.818	0.855	0.870	0.856	0.836	0.840	0.832
	100%	0.840	0.847	0.837	0.814	0.829	0.813	0.851	0.865	0.852	0.836	0.841	0.835
IGM	1%	0.136	0.325	0.069	0.129	0.278	0.278	0.130	0.302	0.059	0.130	0.237	0.061
	2.5%	0.196	0.592	0.165	0.205	0.615	0.180	0.186	0.547	0.151	0.196	0.525	0.164
	5%	0.260	0.612	0.252	0.259	0.706	0.253	0.253	0.703	0.239	0.242	0.683	0.233
	10%	0.302	0.794	0.308	0.300	0.833	0.306	0.321	0.852	0.852	0.323	0.818	0.335
	30%	0.545	0.864	0.595	0.533	0.842	0.576	0.529	0.835	0.577	0.505	0.849	0.553
	50%	0.644	0.780	0.670	0.649	0.776	0.678	0.616	0.801	0.657	0.586	0.735	0.616
	70%	0.647	0.706	0.657	0.660	0.738	0.676	0.637	0.771	0.667	0.634	0.715	0.652
	90%	0.710	0.734	0.712	0.717	0.749	0.720	0.702	0.802	0.722	0.671	0.719	0.678
	100%	0.727	0.751	**0.751**	0.709	0.739	0.711	0.670	0.767	0.688	0.695	0.745	0.705
RF	1%	0.210	0.341	0.170	0.217	0.344	0.176	0.211	0.345	0.171	0.208	0.344	0.166
	2.5%	0.280	0.544	0.269	0.301	0.545	0.291	0.285	0.547	0.274	0.282	0.569	0.278
	5%	0.342	0.568	0.340	0.313	0.563	0.308	0.339	0.569	0.333	0.327	0.568	0.321
	10%	0.397	0.570	0.396	0.385	0.565	0.385	0.394	0.563	0.387	0.392	0.566	0.391
	30%	0.462	0.714	0.452	0.459	0.741	0.460	0.425	0.639	0.440	0.439	0.734	0.447
	50%	0.521	0.864	0.542	0.507	0.850	0.516	0.482	0.826	0.508	0.502	0.854	0.516
	70%	0.533	0.874	0.546	0.546	0.871	0.566	0.489	0.790	0.520	0.530	0.852	0.539
	90%	0.572	0.876	0.571	0.568	0.850	0.583	0.513	0.802	0.547	0.560	0.815	0.573
	100%	0.598	0.856	**0.597**	0.575	0.870	0.593	0.517	0.792	0.547	0.559	0.836	0.588
IDFEC	1%	0.322	0.288	0.259	0.313	0.318	0.251	0.323	0.361	0.287	0.337	0.401	0.306
	2.5%	0.541	0.601	0.512	0.541	0.626	0.518	0.541	0.604	0.530	0.526	0.600	0.506
	5%	0.620	0.644	0.644	0.634	0.648	0.615	0.632	0.692	0.639	0.656	0.663	0.621
	10%	0.727	0.743	0.722	0.706	0.715	0.701	0.741	0.774	0.743	0.714	0.728	0.713
	30%	0.807	0.808	0.803	0.774	0.776	0.771	0.798	0.814	0.798	0.800	0.810	0.799
	50%	0.849	0.851	0.847	0.805	0.806	0.796	0.837	0.850	0.835	0.836	0.838	0.833
	70%	0.845	0.844	0.841	0.837	0.842	0.834	0.842	0.855	0.840	0.849	0.851	0.845
	90%	0.850	0.853	0.849	0.823	0.826	0.820	0.859	0.873	0.861	0.825	0.827	0.822
	100%	0.874	0.877	**0.871**	0.829	0.835	0.824	0.847	0.865	0.848	0.837	0.841	0.834
GI	1%	0.369	0.291	0.294	0.364	0.296	0.293	0.362	0.359	0.318	0.354	0.292	0.281
	2.5%	0.448	0.401	0.387	0.443	0.430	0.385	0.462	0.474	0.474	0.457	0.443	0.281
	5%	0.619	0.671	0.607	0.631	0.738	0.640	0.666	0.738	0.738	0.618	0.697	0.627
	10%	0.717	0.727	0.706	0.712	0.753	0.710	0.717	0.784	0.727	0.703	0.730	0.702
	30%	0.804	0.803	0.799	0.770	0.770	0.770	0.796	0.821	0.802	0.813	0.816	0.810
	50%	0.838	0.838	0.835	0.829	0.828	0.821	0.834	0.842	0.832	0.823	0.829	0.822
	70%	0.839	0.835	0.833	0.824	0.822	0.817	0.867	0.874	0.866	0.849	0.857	0.849
	90%	0.857	0.854	0.852	0.843	0.844	0.838	0.871	0.880	0.871	0.837	0.842	0.834
	100%	0.869	0.870	0.857	0.830	0.837	0.829	0.876	0.889	**0.879**	0.855	0.859	0.854
GT	1%	0.271	0.216	0.201	0.261	0.226	0.199	0.269	0.217	0.203	0.270	0.220	0.203
	2.5%	0.372	0.324	0.306	0.360	0.298	0.292	0.351	0.360	0.304	0.358	0.339	0.305
	5%	0.502	0.649	0.473	0.505	0.675	0.480	0.485	0.648	0.489	0.478	0.624	0.466
	10%	0.627	0.663	0.609	0.651	0.709	0.709	0.626	0.720	0.636	0.628	0.664	0.614
	30%	0.779	0.776	0.769	0.777	0.792	0.771	0.803	0.826	0.809	0.802	0.803	0.798
	50%	0.839	0.841	0.835	0.813	0.814	0.805	0.846	0.856	0.846	0.835	0.839	0.834
	70%	0.842	0.840	0.836	0.838	0.837	0.832	0.843	0.855	0.844	0.846	0.845	0.843
	90%	0.864	0.863	0.859	0.853	0.850	0.848	0.865	0.877	0.867	0.845	0.845	0.841
	100%	0.871	0.874	0.868	0.853	0.854	0.849	0.881	0.893	**0.883**	0.843	0.843	0.840
GSS	1%	0.326	0.307	0.262	0.331	0.293	0.268	0.349	0.353	0.309	0.324	0.386	0.293
	2.5%	0.540	0.625	0.522	0.532	0.644	0.512	0.545	0.632	0.544	0.533	0.650	0.526
	5%	0.663	0.692	0.645	0.646	0.695	0.631	0.657	0.706	0.663	0.659	0.688	0.652
	10%	0.763	0.775	0.762	0.715	0.719	0.703	0.727	0.771	0.738	0.749	0.754	0.744
	30%	0.799	0.795	0.794	0.784	0.782	0.780	0.831	0.850	0.832	0.814	0.819	0.812
	50%	0.850	0.853	0.847	0.818	0.822	0.814	0.851	0.861	0.850	0.837	0.838	0.835
	70%	0.867	0.865	0.864	0.811	0.813	0.805	0.859	0.868	**0.871**	0.835	0.845	0.837
	90%	0.863	0.868	0.862	0.834	0.837	0.830	0.850	0.861	0.851	0.840	0.840	0.837
	100%	0.873	0.871	0.870	0.838	0.844	0.836	0.853	0.870	0.855	0.852	0.855	0.851
NGL	1%	0.350	0.305	0.286	0.343	0.317	0.290	0.342	0.317	0.290	0.349	0.314	0.293
	2.5%	0.445	0.437	0.389	0.455	0.440	0.440	0.436	0.468	0.407	0.457	0.499	0.424
	5%	0.576	0.642	0.557	0.592	0.645	0.645	0.568	0.673	0.574	0.600	0.685	0.600
	10%	0.713	0.746	0.706	0.717	0.752	0.752	0.693	0.753	0.706	0.689	0.741	0.693
	30%	0.815	0.815	0.809	0.772	0.778	0.760	0.788	0.825	0.798	0.789	0.790	0.782

	50%	0.822	0.832	0.826	0.809	0.809	0.803	0.826	0.838	0.825	0.831	0.832	0.828
	70%	0.847	0.839	0.837	0.824	0.824	0.820	0.855	0.863	0.855	0.854	0.856	0.851
	90%	0.861	0.863	0.859	0.842	0.845	0.838	0.846	0.855	0.843	0.860	0.862	0.858
	100%	0.865	0.860	0.859	0.841	0.844	0.837	0.858	0.871	**0.860**	0.858	0.864	0.857
DFS	1%	0.338	0.321	0.285	0.334	0.327	0.286	0.340	0.322	0.290	0.349	0.336	0.299
	2.5%	0.369	0.311	0.300	0.369	0.307	0.301	0.354	0.365	0.307	0.360	0.371	0.303
	5%	0.407	0.458	0.356	0.399	0.487	0.351	0.403	0.524	0.379	0.411	0.604	0.377
	10%	0.493	0.631	0.475	0.485	0.622	0.465	0.473	0.678	0.468	0.490	0.711	0.479
	30%	0.618	0.773	0.611	0.599	0.772	0.591	0.574	0.784	0.592	0.615	0.766	0.607
	50%	0.636	0.784	0.632	0.608	0.779	0.610	0.610	0.800	0.632	0.633	0.789	0.643
	70%	0.651	0.778	0.651	0.658	0.805	0.661	0.648	0.803	0.674	0.663	0.782	0.672
	90%	0.666	0.785	0.662	0.661	0.793	0.669	0.655	0.796	0.677	0.665	0.790	0.676
	100%	0.699	0.790	**0.697**	0.651	0.752	0.646	0.678	0.810	0.692	0.689	0.769	0.695

Table 6.5. Time required to obtain results using the four classifiers with different number of features

Type	**TW schemes**	**Time required (Mean ± SD)**			
		Linear SVC	LR	Bernoulli NB	RF
Un-Supervised	Random	39.799 ± 4.454	68.953 ± 32.912	38.596 ± 4.314	77.878 ± 37.836
	DF	42.713 ± 6.090	63.740 ± 18.362	39.517 ± 4.591	86.948 ± 28.707
	CF	42.631 ± 5.382	64.328 ± 19.192	40.209 ± 4.154	88.519 ± 30.895
	CF-IDF	44.321 ± 6.432	69.930 ± 20.778	41.573 ± 4.357	90.066 ± 30.417
	TF-IDF	43.650 ± 5.883	66.540 ± 17.715	38.938 ± 4.296	83.195 ± 27.746
Supervised	IG	46.935 ± 11.653	66.703 ± 18.210	39.116 ± 3.852	80.272 ± 28.849
	MI	44.419 ± 8.566	64.764 ± 18.940	38.533 ± 4.408	75.253 ± 24.894
	CHI2	43.010 ± 4.972	59.852 ± 29.859	41.844 ± 3.807	81.279 ± 27.914
	OR	40.459 ± 4.217	72.666 ± 29.984	39.016 ± 3.806	78.382 ± 32.596
	GR	40.420 ± 5.209	62.799 ± 17.673	37.912 ± 4.372	81.277 ± 22.971
	CPD	47.464 ± 10.223	86.172 ± 29.285	42.470 ± 6.844	86.728 ± 37.162
	ICF	40.780 ± 4.640	70.953 ± 29.907	39.021 ± 4.206	99.871 ± 52.336
	TF-IDF-ICF	41.125 ± 4.727	67.493 ± 21.915	38.979 ± 4.091	74.426 ± 25.735
	IGM	40.512 ± 4.453	70.320 ± 29.266	38.995 ± 4.034	97.205 ± 49.270
	RF	40.691 ± 4.366	67.979 ± 26.673	38.655 ± 4.350	83.356 ± 35.586
	IDFEC	41.427 ± 5.495	63.011 ± 17.162	38.242 ± 3.620	76.499 ± 25.725
	GI	40.840 ± 5.037	63.193 ± 18.705	38.281 ± 4.240	72.572 ± 23.645
	GT	40.133 ± 4.879	63.273 ± 19.775	37.689 ± 4.223	71.637 ± 23.864
	GSS	40.575 ± 5.351	51.440 ± 23.938	38.042 ± 4.244	73.746 ± 24.463
	NGL	39.792 ± 4.545	64.791 ± 21.096	37.477 ± 4.050	71.479 ± 23.642
	DFS	44.479 ± 4.439	84.440 ± 36.737	37.334 ± 14.443	84.769 ± 32.844
Average		42.199 ± 5.763	67.302 ± 23.718	39.069 ± 4.776	81.684 ± 30.814

The four classifiers provided very close f1-score results, but we can say that linear SVC is slightly superior. Apart from the random selection method, 11 TW schemes reached their peak f1-score with linear SVC, as shown in Table 6.4. Linear SVC reached its peak f1-score of 0.871 using all available features found by IDFEC and IG and reached 0.870 using all available features found by GSS. With Bernoulli NB, five TW schemes reached peak f1-score with it. Bernoulli NB reached his peak f1-score of 0.883 using all available features found by GT, and he reached 0.879 using all available features found by GI and 0.872 using 70% of the available features found by TF-IDF. The remaining two classifiers, LR and RF, gave very close results. Two TW schemes reached their peak performance with LR which peaked at 0.849 using all available features found by GR and GT. MI and NGL reach their peak f1-score with RF which peaks at 0.862, 0.858 with 90% of the available features, respectively. linear

SVC and Bernoulli NB were also the fastest classifiers, as shown in Table 6.5. The results show that Bernoulli NB was the fastest classifier with an average time of 39.069 seconds (with a standard deviation of 4.776), with a very small difference from linear SVC, which came second with an average time of 42.199 seconds (with a standard deviation of 5.763). The other two classifiers, LR, and RF, require a longer time to obtain results. LR required an average of 67.302 seconds (with a standard deviation of 23.718), while RF required the longest time, being the slowest at 81.684 seconds (with a standard deviation of 30.814).

Several TW schemes shows good results with linear SVC, the best performing group being TF-IDF, IG, MI, GSS and IDFEC, which is a mixture of supervised and unsupervised schemes, although the majority were supervised schemes. The worst performing group included RF, CPD and OR, all of which showed very disappointing results. With LR, the supervised TW schemes performed better than the unsupervised schemes, the best performing group included the TF-IDF, IG, MI, NGL and GSS, while the worst-performing group including OR, CPD, and RF, all of which showed very poor results compared to the other schemes. With Bernoulli NB, the best performing group included a mixture of supervised TW schemes, IG, MI, GI, and GSS, and unsupervised schemes, TF-IDF. The worst performing group included RF, IGM, and CPD. The last classifier is RF, the best performing schemes is IG, MI, IDFEC, GSS, and TF-IDF. The worst performing group included IGM, RF, and ICF. The random selection is out of comparison because it does not have a fixed base for TW.

The previous results shows that there are TW schemes that have proven their effectiveness with all classifiers. TF-IDF, IG, MI, GSS, and IDFEC can considered as the best 5 schemes, while RF, ICF, and CPD, in addition to the random selection, can considered as the schemes with weakest f1-score results in our experiments. Between these two groups of schemes, the other schemes showed varied results, as they rose relatively with some classifiers and decreased slightly with others.

6.4.1. Precision and recall

All the results that were mentioned in the so far are f1-score results, this section discusses the results of other evaluation metrics that were used in this thesis which is precision and recall. Most of the TW schemes showed recall slightly higher than the precision, but in some TW schemes the difference between them was slightly larger

than normal. With available features obtained using OR, recall was significantly higher than precision with all four classifiers we used, the difference between them peaking using the best 30% of available features with linear SVC where recall was higher than precision by 0.340. Using CPD, recall was higher than precision with all classifiers where the difference reached to 0.369 using the best 30% of features available with LR. The difference between recall and precision reached its highest value in all experiments using the best 10% of features available using IGM with LR, where the recall value was 0.534 higher than the precision value. These higher recall values indicate that there were very few false negatives and that a classifier is more lenient in the criteria for classifying something as positive. Although recall was slightly higher than precision in most experiments, there were a few minor exceptions in which precision was slightly higher than recall. Using features found by GI, recall and precision were very close, resulting in precision being higher than recall in some cases. Using the best 1% of available features with LR, precision was 0.069 higher than recall. The highest difference in favor of precision was obtained using the best 1% of features available with linear SVC, where precision was 0.078 higher than recall. These higher precision values indicate that there are too few false positives, and the classifier is too strict in the criteria for classifying something as positive.

6.4.2. Filtering FS method summary

Table 6.3 shows that the best TW schemes peaked with a feature count between 1000-4000 features, which is logical, as a small number of features, less than 1000 features, means the absence of many important features, while the large number of features, more than 4000 features, indicates the presence of all features and thus the absence of a great quality of the TW schemes, so, the selection of a limited number of features, but carefully chosen, better than choosing a large number of features that may somehow disrupt the work and accuracy of the algorithm. It is not easy to choose the best classifier or the best TW scheme because the results are constantly changing depending on the type and size of the data, the way the data is pre-processed, and many other factors. According to the results that we discussed in this section, we can say that linear SVC is the best classifier in terms of f1-score results and time to get results, and TF-IDF, IG, MI, GSS, IDFEC are the five best TW schemes in general. These techniques will be used in the next section, where they will be used with the GA in

hybrid FS-GA approach to select fewer features while maintaining the accuracy of the classification and the quality of the results.

6.5. Hybrid FS-GA Approach Results

This section discusses the results of the proposed approach, hybrid FS-GA approach, whereby the best classifier, Linear SVC, and best five TW schemes obtained with filter FS method, TF-IDF, IG, MI, GSS, and IDFEC, will be used with GA. The aim of choosing the best five TW schemes and the best classifier is to reduce the results that need to be discussed and to use only the best methods found by filter FS, which will produce lower results to discuss with higher quality. Reducing the number of TW schemes and classifiers that will be used with hybrid FS-GA approach also contributes to reducing the time required to obtain results, especially within the time constraints and specifications of the devices in which the thesis was conducted. The best subset of features will be selected at 1%, 2.5%, 5%, 10%, 30%, 50%, 70%, and 90% of the original feature size, 1000 features. The goal of selecting these subsets is to outperform the best f1-score results found with filter FS with fewer features. Table 6.6 show precision, recall, and f1-score results of hybrid FS-GA approach and filtering method at different sizes of features.

Table 6.6. Precision, recall, and f1-score results of hybrid FS-GA approach and filtering method at different sizes of features

TW	**Subset size**	**# Feature**	**Proposed Approach**				**Filtering method**
			Precision	Recall	F1-Score	Time (Minutes)	F1-Score
IG	1%	10	0.776	0.656	0.676	204.793	0.298
	2.50%	25	0.796	0.736	0.740	202.337	0.495
	5%	50	0.807	0.758	0.765	203.392	0.663
	10%	100	0.831	0.817	0.818	199.502	0.725
	30%	300	0.898	0.895	0.895*****	203.657	0.797
	50%	500	0.902	0.901	0.900	199.306	0.841
	70%	700	0.921	0.918	0.918	209.840	0.852
	90%	900	0.925	0.922	**0.922**	213.808	0.860
	100%	1000	-	-	-	-	**0.871**
MI	1%	10	0.759	0.624	0.638	198.977	0.307
	2.50%	25	0.772	0.696	0.704	198.399	0.490
	5%	50	0.792	0.748	0.753	200.686	0.635
	10%	100	0.838	0.829	0.828	162.686	0.736
	30%	300	0.904	0.901	0.900*****	203.505	0.797
	50%	500	0.916	0.914	**0.913**	203.300	0.836
	70%	700	0.913	0.910	0.910	207.575	0.840
	90%	900	0.914	0.914	0.913	216.340	0.846
	100%	1000	-	-	-	-	**0.855**
TF-IDF	1%	10	0.705	0.577	0.594	203.947	0.335
	2.50%	25	0.750	0.678	0.691	199.572	0.525
	5%	50	0.769	0.733	0.736	201.737	0.658
	10%	100	0.785	0.775	0.774	199.293	0.743

	30%	300	0.846	0.842	0.841	201.522	0.828
	50%	500	0.881	0.876	0.874*****	201.125	0.839
	70%	700	0.898	0.894	**0.894**	159.835	0.856
	90%	900	0.887	0.883	0.882	161.796	0.859
	100%	1000	-	-	-	-	**0.867**
GSS	1%	10	0.740	0.645	0.657	180.469	0.262
	2.50%	25	0.775	0.711	0.719	198.921	0.522
	5%	50	0.800	0.774	0.772	204.169	0.645
	10%	100	0.830	0.822	0.821	159.235	0.762
	30%	300	0.884	0.885	0.882*****	149.837	0.794
	50%	500	0.912	0.911	0.910	150.603	0.847
	70%	700	0.917	0.918	**0.916**	167.679	0.864
	90%	900	0.906	0.903	0.903	206.732	0.862
	100%	1000	-	-	-	-	**0.870**
IDFEC	1%	10	0.743	0.607	0.631	148.665	0.259
	2.50%	25	0.770	0.691	0.699	206.831	0.512
	5%	50	0.799	0.769	0.771	212.363	0.644
	10%	100	0.804	0.796	0.796	192.777	0.722
	30%	300	0.884	0.880	0.879*****	188.831	0.803
	50%	500	0.907	0.904	0.904	191.103	0.847
	70%	700	0.912	0.913	**0.911**	197.128	0.841
	90%	900	0.906	0.906	0.905	188.247	0.849
	100%	1000	-	-	-	-	**0.871**

Bold: The highest performance found by selected FS method using the features found by TW scheme.
* : The number at which the f1-score of hybrid FS-GA approach began to outperform the best f1-score found using filter FS.

Table 6.6 show the significant superiority of the proposed approach, hybrid FS-GA approach, over filter FS. The proposed approach was able to get better f1-score results than the best f1-score results obtained in filter FS using only 30% of the available features with most of the TW schemes. These results shows the high quality of the proposed approach and its ability to select a small set of features with higher quality than using all available features with filter FS. Using the features found using IG, the f1-score of hybrid FS-GA approach reached 0.895 using only 30% of the available features, while the best f1-score obtained using filter FS was 0.871 using all available features. This superiority using only 30% of the available features is maintained when using the features found using MI, GSS and IDFEC. Using MI, the hybrid FS-GA approach reached f1-score of 0.900 using only 30% of the available features, while the best f1-score obtained using filter FS was 0.855 using all available features. Using GSS and IDFEC as well, the hybrid FS-GA approach reached f1-score of 0.882 and 0.979 using only 30% of the available features, while the best f1-score obtained using filter FS was 0.870 and 0.871 using all available features, respectively. The only case where hybrid FS-GA approach required more than 30% to outperform the highest f1-score obtained by filter FS is using features found by TF-IDF, where hybrid FS-GA approach outperformed the highest filter FS score using 50% of the available features where the f1-score of hybrid FS-GA approach reached 0.874, while the best result obtained using filter FS was 0.867 using all available features. Figure 6.1 shows the

performance of TW schemes that were used with hybrid FS-GA approach with different features sizes more clearly.

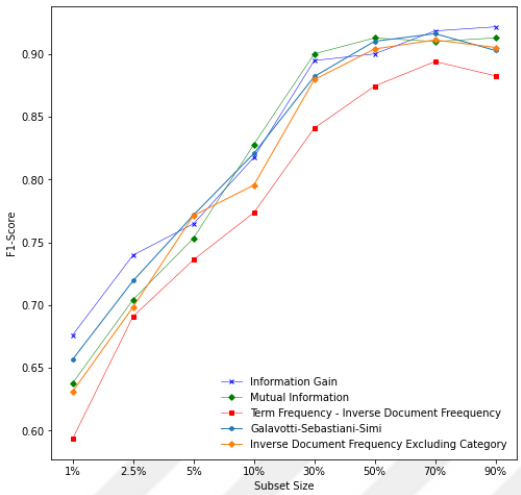

Figure 6.1. Performance of TW schemes within a different set of feature sizes

The results in Table 6.6 showed that the GA performed very well with hybrid FS-GA approach and was characterized by the ability to select features with high quality even when the number of selected features was very few. Hybrid FS-GA approach shows many good results, but in this section two of the most notable results will be discussed. The highest f1-score achieved is 0.922 using 90% of available features found by IG. Figure 6.2(a) shows the highest, average, and lowest f1-score values that were obtained in each generation. The results show the development of the f1-score obtained in each generation, where the highest f1-score obtained in the first generation was about 0.890, the worst f1-score was about 0.840, and the average curve shows that the other f1-score ranged between the best and worst. With the succession of generations, one note the great development of the results of the GA, where the highest f1-score reached 0.922, but what must be looked at carefully is the average values curve, which was gradually approaching the curve of the highest f1-score values, as it was in many stages very close to it, which indicates that the f1-score for most chromosomes were very good, with a few chromosomes having low f1-score values. Figure 6.2(b) shows another important f1-score obtained using 30% of the features found by MI. This result is very important as it is the first time that the proposed approach exceed the 0.900 barrier using a very small number of features, only 300 features, which shows the high ability of hybrid FS-GA approach to select a few features with high performance.

Figure 6.2(b) shows the ascending curve of the best f1-score, where the highest f1-score obtained in the first generation was approximately 0.825 and eventually reached 0.900 with an impressive ascending curve. The mutation value, which was set at 0.1 in the parameters, helped in finding these good results, as the relatively high mutation value in the research helped to obtain different results, especially with the advancement of generations, where all chromosomes become very similar. This value of the mutation rate was accompanied by the creation of some chromosomes whose results were less than expected, as the worst results curve shows.

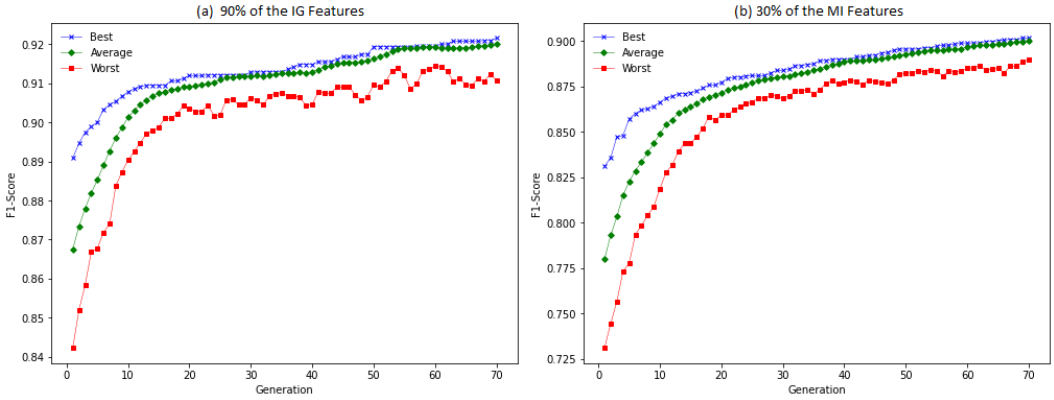

Figure 6.2. Average fitness values (f1-score) found in each generation using hybrid FS-GA approach

6.5.1. Precision and recall

This section discusses the results of other evaluation metrics that were used with hybrid FS-GA approach which is precision and recall. Although the precision and recall showed highly similar results, they showed a distinct pattern that we can observe with all the TW schemes. using a small number of features, we notice that the precision is higher than the recall, but as the number of features increases, we notice that recall gradually approaches precision. Using 1% of available features found by MI, one notice that precision is significantly greater than recall, with a precision of 0.759 and a recall of 0.624, a difference of 0.135. This difference becomes less the more the number of features, as the difference reaches about 0.003 using the best 30% of the available features, as the precision reached 0.904 while the recall was 0.901. When 90% of the available features is selected, it is noted that the difference disappears and the precision and recall become equal, as the value of both was 0.914. This pattern is repeated using the features found by all TW schemes. These results can be explained

that precision is more a measure of quality, while recall can be considered a measure of quantity. Using a small number of features, the approach returns a small number of results, but these few results are of high accuracy since the test data is correctly predicted when compared with the training data, so the precision is high. As the number of features increases gradually, the approach returns a large number of results, even if some of them are incorrectly predicted, so the precision decreases relatively as the recall increases.

6.5.2. Hybrid FS-GA approach summary

Hybrid FS-GA approach showed great efficiency and high stability even when the number of selected features is relatively small, only 1% of the total number of features. These results confirmed the great superiority of the proposed approach over filter FS, where the proposed approach was able to outperform the best results found using filter FS using a much smaller number of features, as it required only 30% of the total number of features to outperform the best filter FS results.

6.6. Hybrid FS-GA Approach With English And Turkish

Hybrid FS-GA approach was evaluated with the English and Turkish to confirm his ability to obtain good results with other languages, not only with Arabic. For English, BBC News dataset were used, which is one of the datasets commonly used in TC research as it is available for use as benchmarks for ML research [152, 153]. The BBC Dataset consists of 2,225 documents from the BBC News website divided into five categories (business, entertainment, politics, sport, and technology) that were published on the website between 2004-2005. For the Turkish, TTC-3600 dataset was used [154]. The dataset consists of 3,600 documents divided into 7 categories (Politics, world, economy, culture, health, sport, and technology) collected by web scraping techniques from seven highly known - and reliable - news agencies in Turkey.

Although some pre-processing was done on the Turkish and English data sets, the pre-processing techniques that were used in this thesis, mentioned in Section 4.4, were applied to both data sets. 1000 documents from both datasets were used in the experiments, noting that the selected documents contain all categories in the original dataset and that the number of documents for each category is equal. Table 6.7 shows

the performance of both datasets, English and Turkish, using filter FS and our proposed approach.

Table 6.7. F1-score of hybrid FS-GA approach and filter FS with English and Turkish datasets

TW	Subset size	# Feature	Turkish		English	
			Filtering method	Proposed approach	Filtering method	Proposed approach
TF-IDF	1%	10	0.402	0.609	0.589	0.888
	2.50%	25	0.480	0.693	0.715	0.900
	5%	50	0.567	0.758	0.828	0.945*****
	10%	100	0.665	0.787	0.905	0.971
	30%	300	0.734	0.851*****	0.915	**0.997**
	50%	500	0.766	**0.879**	0.925	0.996
	70%	700	0.816	0.875	0.924	0.992
	90%	900	0.807	0.878	**0.944**	0.982
	100%	1000	**0.829**	-	0.944	-
IG	1%	10	0.386	0.660	0.747	0.849
	2.50%	25	0.621	0.717	0.852	0.929
	5%	50	0.718	0.753	0.902	0.929
	10%	100	0.757	0.802	0.933	0.972*****
	30%	300	0.812	0.900*****	0.946	0.995
	50%	500	0.810	0.888	0.949	**0.997**
	70%	700	0.813	**0.915**	0.942	0.996
	90%	900	**0.822**	0.915	0.942	0.988
	100%	1000	0.797	-	**0.960**	-
MI	1%	10	0.380	0.693	0.722	0.847
	2.50%	25	0.589	0.754	0.825	0.902
	5%	50	0.739	0.790	0.898	0.945*****
	10%	100	0.754	0.843*****	0.915	0.973
	30%	300	0.819	**0.915**	0.945	**0.995**
	50%	500	0.810	0.915	0.938	0.995
	70%	700	**0.837**	0.915	**0.945**	0.993
	90%	900	0.820	0.898	0.942	0.994
	100%	1000	0.816	-	0.952	-
IDFEC	1%	10	0.436	0.670	0.649	0.867
	2.50%	25	0.562	0.713	0.805	0.899
	5%	50	0.659	0.767	0.884	0.935
	10%	100	0.746	0.831	0.906	0.979*****
	30%	300	0.767	0.876*****	0.932	0.995
	50%	500	0.794	**0.912**	0.954	0.996
	70%	700	0.828	0.903	0.936	**0.998**
	90%	900	**0.840**	0.905	0.936	0.990
	100%	1000	0.820	-	**0.965**	-
GSS	1%	10	0.421	0.683	0.682	0.833
	2.50%	25	0.662	0.714	0.840	0.911
	5%	50	0.739	0.774	0.866	0.939
	10%	100	0.765	0.820*****	0.948	0.978*****
	30%	300	0.802	**0.905**	0.943	0.989
	50%	500	0.809	0.893	0.940	**0.997**
	70%	700	0.804	0.896	0.951	0.991
	90%	900	0.802	0.901	0.939	0.988
	100%	1000	**0.814**	-	**0.955**	-

Bold: The highest performance found by selected FS method using the features found by TW scheme.
* : The number at which the f1-score of hybrid FS-GA approach began to outperform the best f1-score found using filter FS.

With the Turkish dataset, hybrid FS-GA approach showed a clear superiority over the filter FS. Using TF-IDF, IG and IDFEC, hybrid FS-GA approach was able to

outperform the best result found using filter FS using only 30% of the available features. hybrid FS-GA approach provided better results using MI and GSS and was able to outperform the best result found by filter FS using only 10% of the available features. Using filter FS, the Turkish dataset reached a peak f1-score of 0.840 using 90% of available features using IDFEC, while hybrid FS-GA approach reached a peak f1-score of 0.915 using 30% of available features using MI.

Using the English dataset, the results of filter FS increased very clearly, as most of the results exceeded 0.900, however, hybrid FS-GA approach achieved very good results and outperformed the results of filter FS. Using IG, MI, IDFEC, and GSS, hybrid FS-GA approach was able to outperform the best result found using filter FS by using only 10% of the available features. hybrid FS-GA approach gave its best performance using TF-IDF, where it was able to outperform the results found using filter FS using only 5% of the available features. Using filter FS, the English dataset reached a peak f1-score of 0.965 using all available features using IDFEC, while our proposed approach reached a peak f1-score of 0.998 using 70% of the available features using IDFEC as well. Figure 6.3 shows two of the best f1-score given by hybrid FS-GA approach, 0.900 using 30% of the available features found using IG with the Turkish dataset, and 0.979 using 10% of the available features found using IDFEC with the English dataset.

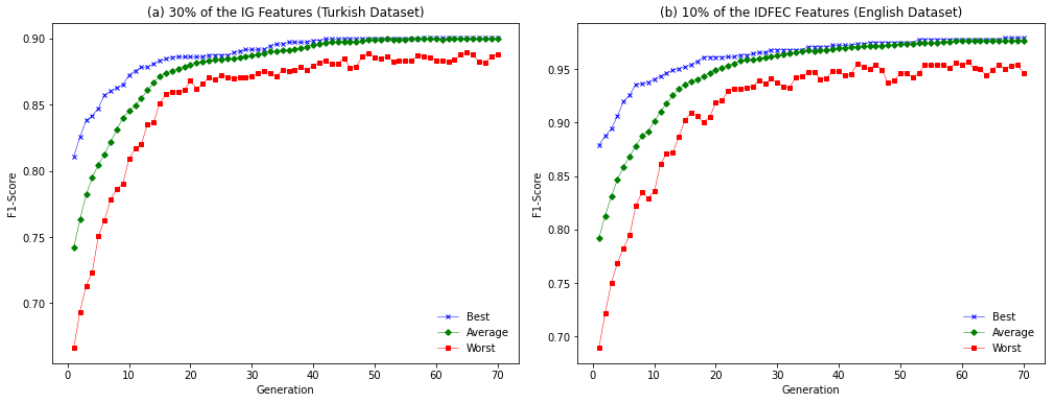

Figure 6.3. Average fitness values (f1-score) found in each generation using hybrid FS-GA approach

7. CONCLUSION

This section presents a summary of the findings and contributions made in this thesis. It also presents some recommendations that may help in the development of future research.

7.1. Discussion

The results, mentioned in section 4.6, shows that linear SVC can consider as the best classifier in general, as it is one of the best classifiers in terms of f1-measure results, and it is the second fastest classifier. As for the TW schemes that are used, one can observe that random selection is not a good solution to reduce the number of features, so other TW schemes must be utilized. The unsupervised TW schemes shows a good result, as TF-IDF is one of the best TW schemes used. Although the results of the other unsupervised schemes are not as good as those of TF-IDF, they also perform good results, and these methods can be preferred especially when pre-classified training data is not available. Supervised TW schemes have shown very good results, as IG, IDFEC, GSS, and MI can be considered as the best four supervised TW scheme. The best classifier (linear SVC classifier) and the five best TW schemes (TF-IDF, IG, MI, IDFEC, and GSS) with filter FS stage are selected to use with the proposed approach, hybrid FS-GA approach. Hybrid FS-GA approach achieved the best f1-score results with the three datasets, Arabic, Turkish and English. With the Arabic dataset, the hybrid FS-GA approach was able to outperform the best f1-score results found using filter FS by using 30%-50% fewer features than the total number of features. With the Turkish dataset, the hybrid FS-GA approach was able to outperform the best f1-score results found using filter FS by using 10%-30% fewer features than the total number of features. The hybrid FS-GA approach was able to achieve its best results with the English dataset, outperforming the best filter FS results using only 5%-10% of the total dataset. In short, the hybrid FS-GA approach is very clearly superior, with three different languages, to the commonly used method for feature selection, the filter FS method. These very good results confirmed the great quality of the proposed approach in all its steps, as we mentioned previously that good results come by applying all the steps, starting from data collection and processing, to obtaining results, with high performance.

7.2. Contributions

This thesis provided a complete explanation of the Arabic in a detailed manner that is unusual for this type of research, where the reader who is not familiar with the Arabic has an overview of it in order to understand the importance of this thesis and the value of the contributions it reached. Due to the lack of high-quality Arabic datasets, this thesis contributes to creating a new high-quality Arabic data set, as the results showed in Section 6.2, and making it publicly available, in all its versions, to researchers and those interested in the Arabic TC. This study also presents a proposed approach, hybrid FS-GA approach, that has proven effective with three languages (Arabic, Turkish and English) and was able to clearly outperform the usual FS method, filter FS. The proposed approach in this study is considered one of the largest systems in terms of the number of techniques and methods used in it, with four data versions, many pre-processing methods, more than 20 TW schemes, four classifiers, GA and other techniques for data collecting and FS.

7.3. Future Work

This section outlines some recommendations that may make a difference in the TC process. Using a larger number of documents or features that were used in this thesis may help improve performance, as some TW techniques, in Section 6.3, showed better f1-score results when using a larger number of features, and using a larger number of documents may also help in find more good features. There are several globalization techniques used with TW schemes including maximum, summation, average, round-robin, and other methods. Changing the globalization method in some TW schemes may produce better results. The development of a new TW scheme, or the combination of more than one TW scheme in one technique, may lead to an increase in the performance, as many of the combined techniques shows better results than working alone, as f1-score results in Section 6.4 shows, such as TF-IDF-ICF, which achieved better results than single ICF results, and CF-IDF which achieved better results than single CF results.

REFERENCES

[1] Internet Growth Summary, "Internet growth statistics 1995 to 2023 - the global village online," *Internetworldstats.com*, 2019. [Online]. Available: https://www.internetworldstats.com/emarketing.htm.

[2] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Text Mining and its Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 81–97.

[3] D. R. J. G. J. Rydning, J. Reinsel, and J. Gantz, The digitization of the world from edge to core. *Framingham: International Data Corporation*. 2018.

[4] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *J. Am. Soc. Inf. Sci. Technol.*, vol. 62, no. 2, pp. 406–418, 2011.

[5] "How many people use Facebook in 2022?," *Backlinko*, 09-Sep-2020. [Online]. Available: https://backlinko.com/facebook-users.

[6] I. Anwar, K. Haider, and S. Iqbal, "Ongoing information search motivation: An influencer of users' approval of Facebook banner adverts," *Journal of Marketing Strategies*, vol. 3, no. 3, pp. 155–170, 2021.

[7] V. Korde, "Text classification and classifiers:A survey," *Int. J. Artif. Intell. Appl.*, vol. 3, no. 2, pp. 85–99, 2012.

[8] J. Yang, Y. Liu, X. Zhu, Z. Liu, and X. Zhang, "A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization," *Inf. Process. Manag.*, vol. 48, no. 4, pp. 741–754, 2012.

[9] C. Manning and H. Schuetze, *Foundations of statistical natural language processing*. London, England: MIT Press, 1999.

[10] S. Alotaibi and M. B. Khan, "Sentiment analysis challenges of informal Arabic," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, 2017.

[11] K. Pearson, "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Lond. Edinb. Dublin Philos. Mag. J. Sci.*, vol. 50, no. 302, pp. 157–175, 1900.

[12] Y. Liu, H. T. Loh, and A. Sun, "Imbalanced text classification: A term weighting approach," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 690–701, 2009.

[13] D. Mladeni'c and M. Grobelnik, "Feature selection for classification based on text hierarchy," in *Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[14] R. Krishnamurthy, "1987-The Cobuild Dictionary (1987)." Unpublished, 1987.

[15] S. Alansary, M. Nagi, and N. Adly, "Building an International Corpus of Arabic (ICA): progress of compilation stage," in *7th international conference on language engineering*, Cairo, Egypt, 2007, pp. 5–6.

[16] E. D. Mindt, "Syntactic evidence for semantic distinctions in English," in *English corpus linguistics*, 2014, pp. 194–208.

[17] C. F. Meyer, "English Corpus Linguistics: An Introduction. Studies in English Language," $, vol. 60, pp. 10011–14211, 2002.

[18] M. J. Meena, K. R. Chandran, and J. M. Brinda, "Integrating swarm intelligence and statistical data for feature selection in text categorization," *Int. J. Comput. Appl.*, vol. 1, no. 11, pp. 16–21, 2010.

[19] Y. Xu, "A comparative study on feature selection in unbalance text classification," in *2012 Fourth International Symposium on Information Science and Engineering*, 2012.
[20] J. Zhang, Q. Wang, Y. Li, D. Li, and Y. Hao, "A method for Chinese text classification based on three-dimensional vector space model," in *International Conference on Computer Science and Service System*, 2012.
[21] H. Suwanwiwat, A. Das, M. Saqib, and U. Pal, "Benchmarked multi-script Thai scene text dataset and its multi-class detection solution," *Multimed. Tools Appl.*, vol. 80, no. 8, pp. 11843–11863, 2021.
[22] A. Cole, D. Graff, and K. Walker, "Arabic Newswire Part 1 Corpus (1-58563-190-6)," in *Linguistic Data Consortium (LDC)*, 2001.
[23] A. Goweder and A. De Roeck, "Assessment of a significant Arabic corpus," in *Arabic NLP Workshop at ACL/EACL*, 2001.
[24] H. K. H. Chantar, New techniques for Arabic document classification (*Doctoral dissertation*). 2013.
[25] M. Hijjawi, Z. Bandar, and K. Crockett, "User's utterance classification using machine learning for Arabic Conversational Agents," in *5th International Conference on Computer Science and Information Technology*, 2013.
[26] R. Belkebir and A. Guessoum, "A hybrid BSO-Chi2-SVM approach to Arabic text categorization," in *ACS International Conference on Computer Systems and Applications (AICCSA)*, 2013.
[27] M. Badry, H. Hassan, H. Bayomi, and H. Oakasha, "QTID: Quran Text Image Dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 3, 2018.
[28] N. A. A.-H. And and M. Prince, "A classification of Arab ethnicity based on face image using deep learning approach," *IEEE Access*, vol. 9, pp. 50755–50766, 2021.
[29] S. Yousfi, S.-A. Berrani, and C. Garcia, "ALIF: A dataset for Arabic embedded text recognition in TV broadcast," in *13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015.
[30] B. Bataineh, "A printed PAW image database of Arabic language for document analysis and recognition," *J. ICT Res. Appl.*, vol. 11, no. 2, p. 200, 2017.
[31] '. Moh and A. Mesleh, "Chi square feature extraction based svms arabic language text categorization system," *Journal of Computer Science*, vol. 3, no. 6, pp. 430–435, 2007.
[32] A. M. Mesleh, "Support vector machines based Arabic language text classification system: Feature selection comparative study," in *Advances in Computer and Information Sciences and Engineering*, Dordrecht: Springer Netherlands, 2008, pp. 11–16.
[33] H. K. Chantar and D. W. Corne, "Feature subset selection for Arabic document categorization using BPSO-KNN," in *Third World Congress on Nature and Biologically Inspired Computing*, 2011.
[34] G. Raho, R. Al-Shalabi, G. Kanaan, and A. Nassar, "Different classification algorithms based on Arabic text classification: Feature selection comparative study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 2, 2015.
[35] 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 - Semitic '04*, 2004.

[36] A. M. El-Halees, “Arabic text classification using maximum entropy,” *IUG Journal of Natural Studies*, no. 1, 2015.
[37] G. Kanaan, R. Al-Shalabi, and A. Al-Akhras, “kNN Arabic text categorization using IG feature selection,” *Proceedings of The 4th International Multiconference on Computer Science and Information Technology*, vol. 4, pp. 5–7, 2006.
[38] 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, no. 1, pp. 1–19, 2006.
[39] M. A. Basir, Y. Yusof, and M. S. Hussin, “Optimization of attribute selection model using bio-inspired algorithms,” *J. ICT*, 2018.
[40] W. Deng, R. Yao, M. Sun, H. Zhao, Y. Luo, and C. Dong, “Study on a novel fault diagnosis method based on integrating EMD, fuzzy entropy, improved PSO and SVM,” *J. Vibroengineering*, vol. 19, no. 4, pp. 2562–2577, 2017.
[41] S. Mirjalili, I. Aljarah, M. Mafarja, A. A. Heidari, and H. Faris, “Grey wolf optimizer: Theory, literature review, and application in computational fluid dynamics problems,” in *Nature-Inspired Optimizers, Cham: Springer International Publishing*, 2020, pp. 87–105.
[42] M. Hijazi, A. Zeki, and A. Ismail, “Arabic Text Classification Using Hybrid Feature Selection Method Using Chi-Square Binary Artificial Bee Colony Algorithm,” *Computer Science*, vol. 16, no. 1, pp. 213–228, 2021.
[43] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, “Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification,” *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12201–12220, 2020.
[44] S. Alqurashi, A. Alhindi, and E. Alanazi, “Large Arabic Twitter Dataset on COVID-19,” *arXiv [cs.SI]*, 2020.
[45] G. Dhaou and G. Lejeune, “Comparison between Voting Classifier and Deep Learning methods for Arabic Dialect Identification,” in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 2020, pp. 243–249.
[46] A. Mohammed and R. Kora, “Deep learning approaches for Arabic sentiment analysis,” *Soc. Netw. Anal. Min.*, vol. 9, no. 1, 2019.
[47] A. S. Alshuaibi, F. Mohd Shamsudin, and M. S. I. Alshuaibi, *Internet misuse at work in Jordan: Challenges and implications*. 2015.
[48] “Top ten Internet languages in the world - Internet statistics,” *Internetworldstats.com*, 2017. [Online]. Available: https://www.internetworldstats.com/stats7.htm.
[49] “Domain Name Industry Brief (DNIB) - verisign,” *Verisign.com*. [Online]. Available: https://www.verisign.com/en_US/domain-names/dnib/index.xhtml.
[50] “Measuring digital development: Facts and figures 2021,” *International Telecommunication Union (ITU)*. Retrieved, vol. 16, 2021.
[51] A. Farghaly and K. Shaalan, “Arabic natural language processing: Challenges and solutions,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 8, no. 4, pp. 1–22, 2009.
[52] “World Internet Users statistics and 2023 world Population stats,” *Internetworldstats.com*, 2022. [Online]. Available: https://www.internetworldstats.com/stats.htm.

[53] M. G. Peletz, “Hegemonic Muslim masculinities and their others: Perspectives from South and Southeast Asia,” *Comp. Stud. Soc. Hist.*, vol. 63, no. 3, pp. 534–565, 2021.
[54] M. K. Nydell, Understanding Arabs: A contemporary guide to Arab society. *Nicholas Brealey*, 2018.
[55] Alaraby.co.uk. [Online]. Available: https://www.alaraby.co.uk/-أعداد-المسيحيين-العرب-متحف-الأرقام-والتحولات-العاصفة-2/8.
[56] C. Bux, E. Varese, V. Amicarelli, and M. Lombardi, “Halal food sustainability between certification and blockchain: A review,” Sustainability, vol. 14, no. 4, p. 2152, 2022.
[57] R. A. Lulu, S. M. H. A. Racman, and M. Gubat-Manabilang, “Deciphering the Stereotyping of Arabs, their language and religion by the Bruneians,” *Firdaus Journal*, vol. 2, no. 1, pp. 15–23, 2022.
[58] N. Ostler, *Empires of the word: A language history of the world*. London, England: HarperPerennial, 2010.
[59] M. S. H. Mukhopadhyaya, *A Grammar Of The Classical Arabic Language : Translated And Compiled From The Works Of The Most Approved Native Or Naturalized Authorities, Vol.4.* Gyan Publishing House, 2003.
[60] A. Soudi, G. Neumann, and A. van den Bosch, “Arabic computational morphology: Knowledge-based and empirical methods,” in *Text, Speech and Language Technology*, Dordrecht: Springer Netherlands, 2007, pp. 3–14.
[61] M. A. Attia, “An ambiguity-controlled morphological analyzer for modern standard Arabic modeling finite state networks,” in *Proceedings of the International Conference on the Challenge of Arabic for NLP/MT*, 2006.
[62] N. AbdulJaleel and L. S. Larkey, “Statistical transliteration for english-arabic cross language information retrieval,” in *Proceedings of the twelfth international conference on Information and knowledge management - CIKM ’03*, 2003.
[63] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
[64] A. I. Kadhim, “Survey on supervised machine learning techniques for automatic text classification,” *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 273–292, 2019.
[65] G. Forman, “Feature selection for text classification,” in *Computational Methods of Feature Selection, Chapman and Hall/CRC*, 2007, pp. 257–276.
[66] R. Feldman and J. Sanger, The text mining handbook: advanced approaches in analyzing unstructured data. *Cambridge university press*, 2007.
[67] L. S. Larkey, “A patent search and classification system,” in *Proceedings of the fourth ACM conference on Digital libraries*, 1999.
[68] O. Ferret, “Improving text segmentation by combining endogenous and exogenous methods,” in *Proceedings of the International Conference RANLP-2009*, 2009, pp. 88–93.
[69] Q. Wu, Y. Sano, H. Takayasu, and M. Takayasu, “Classification of endogenous and exogenous bursts in collective emotions based on Weibo comments during COVID-19,” *Sci. Rep.*, vol. 12, no. 1, p. 3120, 2022.
[70] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani, “On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14),* 2014, pp. 810–817.

[71] G. G. A. Malhotra, “Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example),” in *International Journal of Computer Applications (0975 – 8887)*, 2015.
[72] A. Ameed et al., “Arabic light stemmer: A new enhanced approach,” in *The Second International Conference on Innovations in Information Technology (IIT’05,* 2005, pp. 1–9.
[73] S. Bahassine, M. Kissi, and A. Madani, “New stemming for Arabic text classification using feature selection and decision trees,” in *Proceedings of the 5th International Conference on Arabic Language Processing*, 2014, pp. 26–27.
[74] D. Agnihotri, K. Verma, and P. Tripathi, “Variable Global Feature Selection Scheme for automatic classification of text documents,” *Expert Syst. Appl.,* vol. 81, pp. 268–281, 2017.
[75] B. Tang, S. Kay, and H. He, “Toward optimal feature selection in naive Bayes for text categorization,” *IEEE Trans. Knowl. Data Eng.,* vol. 28, no. 9, pp. 2508–2521, 2016.
[76] X. Wang, L. Yang, D. Wang, and L. Zhen, “Improved TF-IDF keyword extraction algorithm,” *Computer Science & Application*, vol. 3, no. 1, pp. 64–68, 2013.
[77] Y. Gu and X. Gu, “A supervised term weighting scheme for multi-class text categorization,” in *Intelligent Computing Methodologies*, Cham: Springer International Publishing, 2017, pp. 436–447.
[78] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” In *Icml,* vol. 97, no. 412, 1997.
[79] J. Chen, H. Huang, S. Tian, and Y. Qu, “Feature selection for text classification with Naïve Bayes,” *Expert Syst. Appl.,* vol. 36, no. 3, pp. 5432–5435, 2009.
[80] A. K. Uysal, “An improved global feature selection scheme for text classification,” *Expert Syst. Appl.,* vol. 43, pp. 82–92, 2016.
[81] D. Agnihotri, K. Verma, P. Tripathi, and B. K. Singh, “Soft voting technique to improve the performance of global filter based feature selection in text corpus,” *Appl. Intell.,* vol. 49, no. 4, pp. 1597–1619, 2019.
[82] M. Lan, C. L. Tan, J. Su, and Y. Lu, “Supervised and traditional term weighting methods for automatic text categorization,” *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 31, no. 4, pp. 721–735, 2009.
[83] Hall, M. A. and Smith, L. A., “Practical feature subset selection for machine learning,” in *Computer Science ’98 Proceedings of the 21st Australasian Computer Science Conference ACSC’98*, 1998, pp. 181–191.
[84] K. Chen, Z. Zhang, J. Long, and H. Zhang, “Turning from TF-IDF to TF-IGM for term weighting in text classification,” *Expert Syst. Appl.,* vol. 66, pp. 245–260, 2016.
[85] J. Polpinij and K. Namee, “Improving of imbalanced data in multiclass classification for sentiment analysis using supervised term weighting,” in *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, 2021.
[86] M. Lan, C. L. Tan, and H. B. Low, “Proposing a new term weighting scheme for text categorization,” *AAAI*, vol. 6, pp. 763–768, 2006.
[87] G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori, “A comparison of term weighting schemes for text classification and sentiment analysis with a

supervised variant of tf. idf," in Int*ernational Conference on Data Management Technologies and Applications*, Cham: Springer, 2015, pp. 39–58.
[88] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Syst.* Appl., vol. 33, no. 1, pp. 1–5, 2007.
[89] K. Kim and S. Y. Zzang, "Trigonometric comparison measure: A feature selection method for text categorization," *Data Knowl. Eng.*, vol. 119, pp. 1–21, 2019.
[90] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowl. Based Syst.*, vol. 36, pp. 226–235, 2012.
[91] 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*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 59–68.
[92] A. Adel, N. Omar, and A. Al-Shabi, "A comparative study of combined feature selection methods for Arabic text classification," *J. Comput. Sci.*, vol. 10, no. 11, pp. 2232–2239, 2014.
[93] H. Liu and H. Motoda, Eds., *Computational methods of feature selection*. Philadelphia, PA: Chapman & Hall/CRC, 2007.
[94] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," *SIGIR Forum*, vol. 31, no. SI, pp. 67–73, 1997.
[95] M. E. Ruiz and P. Srinivasan, "Hierarchical neural networks for text categorization (poster abstract)," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.
[96] M. Simeon and R. Hilderman, "Categorical proportional difference: A feature selection method for text categorization," *Proceedings of the 7th Australasian Data Mining Conference*, vol. 87, pp. 201–208, 2008.
[97] H. Wu and G. Salton, "A comparison of search term weighting: term relevance vs. inverse document frequency," pp. 30–39, 1981.
[98] S. Parameswaran and K. Rainey, "Vessel classification in overhead satellite imagery using weighted" bag of visual words"," *Automatic target recognition XXV*, vol. 9476, pp. 83–89, 2015.
[99] C. Deisy, M. Gowri, S. Baskar, S. M. A. Kalaiarasi, and N. Ramraj, "A novel term weighting scheme midf for text categorization," *Journal of Engineering Science and Technology*, vol. 5, no. 1, pp. 94–107, 2010.
[100] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," in *Journal of Machine Learning Research 3 (2003) 1289-1305*, 2003.
[101] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, K. Tzeras, and G. Knorz, "AIR/X-a Rule Based Multistage Indexing System for Large Subject Fields," *RIAO*, vol. 91, pp. 606–623, 1991.
[102] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," in *Arxiv Computation and Language*, 2017.

[103] A. S. Vieira, L. Borrajo, and E. L. Iglesias, “Improving the text classification using clustering and a novel HMM to reduce the dimensionality,” *Computer methods and programs in biomedicine*, vol. 136, pp. 119–130, 2016.
[104] P. Norvig and S. Russell, *Artificial Intelligence: A Modern Approach*. Pearson Higher Ed, 2011.
[105] Y. Y. Liu, M. Yang, M. Ramsay, X. S. Li, and J. W. Coid, “A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending,” *J. Quant. Criminol.*, vol. 27, no. 4, pp. 547–573, 2011.
[106] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” *arXiv [cs.CL]*, 2002.
[107] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, “Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification,” in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, 2019.
[108] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes-which naive bayes?,” In *CEAS*, vol. 17, pp. 28–69, 2006.
[109] S. R. Sain and V. N. Vapnik, “The nature of statistical learning theory,” *Technometrics*, vol. 38, no. 4, p. 409, 1996.
[110] T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features,” in *Machine Learning: ECML-98*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142.
[111] X. Luo, W. Ohyama, T. Wakabayashi, and F. Kimura, “Impact of word segmentation errors on automatic Chinese text classification,” in *10th IAPR International Workshop on Document Analysis Systems*, 2012.
[112] A. Alahmadi, “Automatic text classification using bag of words and bag of concepts based representations,” Univ. of Limerick, Ireland, 2016.
[113] W. Jiang, D.-S. Huang, and S. Li, “Random walk-based solution to triple level stochastic point location problem,” *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1438–1451, 2016.
[114] “Sklearn.Svm.LinearSVC,” *scikit-learn*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.
[115] B. Mahesh, “Machine learning algorithms-a review,” *International Journal of Science and Research*, vol. 9, no. IJSR, pp. 381–386, 2020.
[116] P. H. Swain and H. Hauska, “The decision tree classifier: Design and potential,” *IEEE Trans. Geosci. Electron.*, vol. 15, no. 3, pp. 142–147, 1977.
[117] C. Z. Janikow, “Fuzzy decision trees: issues and methods,” *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 28, no. 1, pp. 1–14, 1998.
[118] B. A. Wahiba and B. E. F. Ahmed, “New fuzzy decision tree model for text classification,” in *Advances in Intelligent Systems and Computing*, Cham: Springer International Publishing, 2016, pp. 309–320.
[119] J. Ababneh, “Application of naïve Bayes, decision tree, and K-nearest neighbors for automated text classification,” *Mod. Appl. Sci.*, vol. 13, no. 11, p. 31, 2019.
[120] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Precioso, and P. Lloret, “Short text classification using semantic random forest,” in *Data Warehousing and Knowledge Discovery*, Cham: Springer International Publishing, 2014, pp. 288–299.

[121] N. Shafiabady, L. H. Lee, R. Rajkumar, V. P. Kallimani, N. A. Akram, and D. Isa, "Using unsupervised clustering approach to train the Support Vector Machine for text classification," *Neurocomputing*, vol. 211, pp. 4–10, 2016.
[122] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.
[123] S. K. Dwivedi and C. Arya, "Automatic text classification in information retrieval: A survey," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–6.
[124] B. Altınel and M. C. Ganiz, "A new hybrid semi-supervised algorithm for text classification with class-based semantics," *Knowl. Based Syst.*, vol. 108, pp. 50–64, 2016.
[125] C. A. Floudas *et al.*, *Handbook of test problems in local and global optimization*, vol. 33. Springer Science & Business Media, 2013.
[126] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM J. Comput.*, vol. 2, no. 2, pp. 88–105, 1973.
[127] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, 1996.
[128] A. Tuncer and M. Yildirim, "Chromosome coding methods in genetic algorithm for path planning of mobile robots," in *Computer and Information Sciences II*, London: Springer London, 2011, pp. 377–383.
[129] J. G. Speight, *Natural gas: a basic handbook*. Gulf Professional Publishing, 2018.
[130] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Comput. Electr. Eng.*, vol. 38, no. 6, pp. 1564–1572, 2012.
[131] K. Deb, "Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction," in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, London: Springer London, 2011, pp. 3–34.
[132] Y. Yun and M. Gen, "Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics," *Fuzzy optimization and decision making*, vol. 2, pp. 161–175, 2003.
[133] A. Pietramala, V. L. Policicchio, P. Rullo, and I. Sidhu, "A genetic algorithm for text classification rule induction," in *Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 188–203.
[134] N. Suguna and K. Thanushkodi, "An improved k-nearest neighbor classification using genetic algorithm," I*nternational Journal of Computer Science Issues*, vol. 7, no. 2, pp. 18–21, 2010.
[135] K. Deep, A. Nagar, and M. Pant, *Proceedings of the International Conference on Soft Computing for Problem Solving*, vol. 2. Springer Science & Business Media, 2011.
[136] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*. John Wiley & Sons, 2004.
[137] G. Desjardins, R. Godin, and R. Proulx, "A genetic algorithm for text mining," *WIT Transactions on Information and Communication Technologies*, 2005.

[138] S. Mokhatab, J. Y. Mak, J. Valappil, and D. Wood, *Handbook of liquefied natural gas*. Gulf Professional Publishing, 2013.
[139] J. Carr, "An introduction to genetic algorithms," *Senior Project*, vol. 1, 2014.
[140] H. Zou and T. Hastie, *Model building and feature selection with genomic data. Computational Methods of Feature Selection*. Chapman & Hall/CRC, 2007.
[141] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," *Third annual symposium on document analysis and information retrieval*, vol. 33, pp. 81–93, 1994.
[142] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing Microarray Gene Expression Data: McLachlan/microarray gene expression*. Nashville, TN: John Wiley & Sons, 2004.
[143] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?," *Comput. Stat.*, vol. 36, no. 3, pp. 2009–2031, 2021.
[144] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, and J. Hu, "Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation," *Computational Materials Science*, vol. 171, 2020.
[145] M. Kuhn and K. Johnson, *Applied predictive modeling*, vol. 26. New York: Springer, 2013.
[146] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information retrieval*, vol. 1, no. 1, pp. 69–90, 1999.
[147] A. Kumar, *Confidence Intervals Formula, Examples*. 2023.
[148] M. O. Ogunlana et al., "Pattern of rape and femicide during COVID-19 lockdown: content and discourse analysis of digital media reports in Nigeria," *J. Gend. Stud.*, pp. 1–14, 2021.
[149] A. H. Al-Dulaimi, "Ultimate Arabic News Dataset," *Efficient arabic text classification using feature selection techniques and genetic algorithm*. 2022.
[150] A. Al-Dulaimi and M. Okkalioglu, "Efficient Arabic text classification using feature selection techniques and genetic algorithm," in *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, 2022.
[151] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Syst. Appl.*, vol. 49, pp. 31–47, 2016.
[152] U. C. D. Derek Greene, "BBC datasets," *Ucd.ie*. [Online]. Available: http://mlg.ucd.ie/datasets/bbc.html.
[153] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd international conference on Machine learning - ICML '06,* 2006.
[154] D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, and E. Borandag, "TTC-3600: A new benchmark dataset for Turkish text categorization," *J. Inf.* Sci., vol. 43, no. 2, pp. 174–185, 2017.

RESUME

He completed his primary and secondary education in Baghdad/Iraq. In 2018 he graduated from the Department of Computer Engineering from Al-Mamoun University College in Baghdad/Iraq. Student at Yalova University Graduate Institute - Department of Computer Engineering since 2020.

PUBLICATIONS AND WORKS DERIVED FROM THE THESIS

[1] A. Al-Dulaimi and M. Okkalioglu, "Efficient Arabic text classification using feature selection techniques and genetic algorithm," in *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, 2022.

[2] A. H. Al-Dulaimi, "Ultimate Arabic News Dataset," *Efficient arabic text classification using feature selection techniques and genetic algorithm*. 2022.

OTHER PUBLICATIONS AND WORKS

[1] G. Guclu and A. Al-Dulaimi, "Estimating and analyzing the spread of Covid-19 in Turkey using Long Short-Term Memory," in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021.