

**SENTIMENT ANALYSIS FOR SPORTS FANATICISM IN ARABIC
SOCIAL MEDIA TEXT**

BY

Mohammed Mughahed Alqmase

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

April 2019

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by Mohammed Mugahed Alqmase under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE**.



Dr. Hamoud Aljamaan
Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies

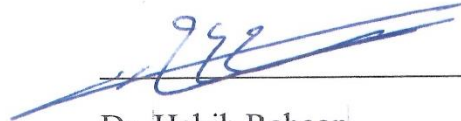


1/1/19

Date



Dr. Husni Al-Muhtaseb
(Advisor)



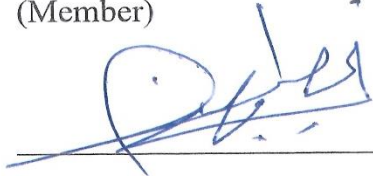
Dr. Habib Rabaan
(Co-Advisor)



Dr. Mohammad Alshayeb
(Member)



Dr. Moataz Ahmed
(Member)



Dr. Wasfi Al-Khatib
(Member)

© Mohammed Alqmase

2019

I declare that the work in this Master thesis is the result of my own work which has not been submitted by others elsewhere for any other degree or qualification.

ACKNOWLEDGMENTS

Praise is to Allah, the Almighty for having guided me in my life. Thanks and sincere gratefulness to my supervisor Dr. Husni Al-Muhtasab where this research would never have been conducted without his valuable suggestions, assistance, encouragement, guidance, advice, support and helpful feedback.

I am greatly indebted to my thesis committee members: Dr. Habib Rabaan, Dr. Moataz Ahmed, Dr. Mohammad Alshayeb and Dr. Wasfi Al-Khatib for their interest, encouragement, recommendations, support and valuable feedback. In addition, I would like to extend my thanks to the academic staff of the Faculty of Information and Computer Science (ICS) who helped me during my master's study and taught me helpful courses. I would like to acknowledge the support of KFUPM university for providing me a suitable environment to complete my master's degree.

Finally, I would also like to express my thanks to my friends and family for their support during my course studies and during my thesis work. |

TABLE OF CONTENTS

ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	IX
LIST OF FIGURES.....	XII
LIST OF ABBREVIATIONS.....	XIV
ABSTRACT.....	XV
ملخص الرسالة	XVII
1 CHAPTER INTRODUCTION.....	1
1.1 Problem Statement	2
1.2 Proposed Solution	3
1.3 Research Goal.....	3
1.4 Research Objectives.....	3
1.5 Research Questions	4
1.6 Research Motivation	4
1.7 Research Scope.....	5
1.8 Research Outcomes	6
1.9 Research Challenges	6
1.10 Research Structure.....	7
2 CHAPTER LITERATURE REVIEW	9
2.1 Sport Fanaticism	9
2.2 Building Resources.....	10

2.2.1	Building Sentiment Lexicons	11
2.2.2	Building Annotated Corpora.....	27
2.3	Feature Engineering.....	35
2.4	Classification.....	40
3	CHAPTER RESEARCH METHODOLOGY.....	48
4	CHAPTER SPORT FANATICISM FORMALIZATION.....	55
4.1	Formalizing Sport Fanaticism into Indicators/Features.....	56
4.1.1	Fanatic Indicators.....	58
4.1.2	Non-fanatic Indicators.....	69
5	CHAPTER DATA COLLECTION.....	77
6	CHAPTER DATA PREPROCESSING.....	89
6.1	Cleaning.....	90
6.2	Normalization.....	90
6.3	Stemming and lemmatization.....	93
7	CHAPTER BUILDING SENTIMENT RESOURCES.....	99
7.1	Building Fanatic-Lexicon Manually.....	99
7.2	Building Fanatic-Lexicon Automatically.....	103
7.2.1	Generate Lexicon Using PPMI.....	104
7.2.2	Generate Lexicon Using the proposed-TFICF.....	113
7.3	Building Fanatic-corpus Automatically.....	117
7.4	Building Fanatic-corpus Semi-Automatically.....	120
8	CHAPTER FEATURE ENGINEERING.....	126
8.1	Feature Identification and Extraction.....	129
8.1.1	Phonological Features.....	129
8.1.2	Morphological features.....	133
8.1.3	Syntactic Features.....	134
8.1.4	Semantic Features.....	137
8.1.5	Stylistic Features.....	139

8.2	Feature Reduction	140
9	CHAPTER CLASSIFICATION AND EVALUATIONS	143
9.1	Experimental Setup	143
9.1.1	Training dataset	143
9.1.2	Test dataset	144
9.1.3	Tools	144
9.1.4	Classifiers	145
9.1.5	Evaluation Metrics	146
9.2	Rule-Based Classification Approach	150
9.2.1	Classifications Using a Manual Lexicon	150
9.2.2	Classifications Using Automatically Generated Lexicons	150
9.3	Learning-Based Classification Approach	159
9.3.1	Stylistic Features Impact	160
9.3.2	Phonological Features Impact	162
9.3.3	The Impact of Bag-of-Stems unigrams and bigrams Features	163
9.3.4	Emojis Unigrams and Bigrams Impact	168
9.3.5	Punctuation, hashtag, emojis and stems	170
9.3.6	Syntactical Features	171
9.4	Hybrid Classification Approach	175
9.5	Summary	181
10	CHAPTER DISCUSSION AND CONCLUSION	184
10.1	Conclusion	185
10.2	Developed Tools	186
10.2.1	A Fanatic/ Non-Fanatic Classifier	187
10.2.2	An Annotation Tool	187
10.2.3	A Cleaner	188
10.2.4	A Normalizer	188
10.2.5	A Feature Extraction Tool	189
10.2.6	Other Tools	189
10.3	Limitations and Future Work	189
	REFERENCES	192
	VITAE	201

LIST OF TABLES

Table 1. Comparing Sentiment Lexicons.....	22
Table 2. Comparing Annotated Corpora.....	32
Table 3. Some words and phrases for the context of threat	59
Table 4. Some words and phrases for the context of insult	60
Table 5. Some words and phrases for the context of vengeance	61
Table 6. Some words and phrases for the context “instigating fans”	62
Table 7. Some words and phrases for accusation context	63
Table 8. Some words and phrases of promoting malpractices context.....	63
Table 9. Some words and phrases of animosity and racism context	64
Table 10. Some words and phrases of contempt context.....	65
Table 11. Some words and phrases of obduracy.....	66
Table 12. Some words and phrases of extreme-sadness	67
Table 13. Some words and phrases of extreme happiness with prejudice context	68
Table 14. Some words and phrases of indignation	68
Table 15. Some words and phrases of sportsmanship	69
Table 16. Some words and phrases of containment context.....	70
Table 17. Some words and phrases of responsibility.....	71
Table 18. Some words and phrases of tolerance.....	72
Table 19. Some words and phrases of cooperation.....	72
Table 20. Some words and phrases of respecting rules	73
Table 21. Some words and phrases of respecting others’ abilities	74
Table 22. Some words and phrases of respecting others’ beliefs and opinions.....	74
Table 23. Some words and phrases of broad-minded context	75
Table 24. Some words and phrases of suggestion context.....	76
Table 25. Abstract summary about the collected data	81
Table 26. Dataset statistics summary.....	84
Table 27. Frequencies of words' unigrams	85
Table 28. Frequencies of words' bigrams	85
Table 29. Frequencies of words' trigrams	86
Table 30. Frequencies of characters' unigrams.....	86
Table 31. Frequencies of emojis' and related-symbols' unigrams	87
Table 32. Frequencies of emojis' and related-symbols' bigrams	87
Table 33. Frequencies of emojis' and related-symbols' trigrams	87
Table 34. Data cleaning examples	90
Table 35. The Effect of non-normalized tweets on the size of features	92
Table 36. The Effect of normalized tweets on the size of features.....	92
Table 37. Data normalization examples.....	93
Table 38. Example of over-preprocessing and under-preprocessing	95

Table 39. Examining the impact of five stemmers	97
Table 40: Comparing eight stemmers	98
Table 41: Keyword for each context.....	100
Table 42: Sport domain keywords	101
Table 43. Examples of Annotated dataset using the manual-fanatic-lexicon.....	108
Table 44. An example of term-context matrix.....	109
Table 45. Example for computing the PPMI equation's terms	110
Table 46. The output of applying (PMI) on the example of Table 44	111
Table 47. The output of applying (PPMI) on the example of Table 44.....	112
Table 48. Generated fanatic-lexicon for ngrams text.....	112
Table 49. Generated fanatic-lexicon for Emojis and related-symbol	112
Table 50. Applying TF on the term-context matrix of Table 44	115
Table 51. Row-wise normalization for (TF) values.....	116
Table 52. The result of the proposed TFICF.....	117
Table 53: Test dataset Statistics	121
Table 54. Evaluation results of the manual fanatic-lexicon.....	122
Table 55. Contents comparison of the manual fanatic-lexicon evaluation.....	122
Table 56. Examples of ambiguous Arabic terms	122
Table 57. Examples of incorrect annotations due to negation	123
Table 58: Under-sampling output distribution for tweets without emojis	124
Table 59. Under-sampling output distribution for tweets with emojis	125
Table 60. Proposed phonological features	130
Table 61. Identified syntactic features	135
Table 62. Identified semantic features	138
Table 63. Identified stylistic features.....	139
Table 64. Confusion Matrix for domain-problem.....	147
Table 65. The impact of different techniques for under-sampling on preformance	154
Table 66. The performance of seven fanatic-lexicons generated by PPMI method	156
Table 67. The performance of seven fanatic-lexicons generated by the proposed- TFICF method.....	158
Table 68. The impact of the proposed stylistic features	161
Table 69. The impact of the proposed phonological features	162
Table 70. Classifications using Unigrams Bag-Of-Stems Feature	164
Table 71. Classifications of bigrams bag-of-stems feature.....	166
Table 72. Combining unigrams and bigrams bag-of-stems feature Classifications	167
Table 73. Combining unigrams and bigrams bag-of-emojis feature Classifications.....	169
Table 74. Combining punctuation, hashtag, emojis and stems features Classifications.	170
Table 75. The impact of the proposed syntactical features.....	172
Table 76. The impact of transient words as bag-of-words feature representation.....	174

Table 77. Classifications using the proposed 16 features extracted from PPMI
lexicons 177

Table 78. T Classifications using the proposed 16 features extracted from TFICF
lexicons 178

Table 79. Classifications using the proposed 32 features extracted from both PPMI
and TFICF lexicons..... 180

LIST OF FIGURES

Figure 1. Senti-lexicon constructed approaches	14
Figure 2. Classification Approaches	41
Figure 3. Research Methodology	49
Figure 4. Definition of Fanatic-Text.....	57
Figure 5. Definition of Non-Fanatic-Text.....	57
Figure 6. Overview for Sport Fanaticism Formalization (SFF).....	58
Figure 7. An example of threat context	59
Figure 8. Example of insult context.....	60
Figure 9. Example of vengeance context.....	61
Figure 10. Example of the context “instigating fans”.....	62
Figure 11. An example of accusation context.....	63
Figure 12. An example of the promoting malpractices context.....	64
Figure 13. An example of animosity and racism	64
Figure 14. An example of contempt	65
Figure 15. An example of obduracy	66
Figure 16. An example of extreme-sadness context	67
Figure 17. An example of extreme happiness with prejudice.....	68
Figure 18. An example of indignation	68
Figure 19. An example of sportsmanship	70
Figure 20. An example of containment.....	70
Figure 21. An example of responsibility context.....	71
Figure 22. An example of tolerance.....	72
Figure 23. An example of cooperation	72
Figure 24. An example of respecting rules	73
Figure 25. An example of respecting others’ abilities	74
Figure 26. An example of respecting others’ beliefs and opinions.....	75
Figure 27. An example of broad-minded context	75
Figure 28. An example of suggestion context	76
Figure 29. Example of one of the received objects from Twitter APIs	79
Figure 30. Search string applied on Twitter APIs.....	80
Figure 31. Time distribution of collected data.....	81
Figure 32. Frequencies of bigrams terms of the search string	82
Figure 33. Unigrams term frequencies of the search string	83
Figure 34. Frequencies of Saudi football teams' in the collected dataset	84
Figure 35. A preprocessed example.....	89
Figure 36. Twitter's search box.....	101
Figure 37. Web app to simplify adding new context-related words and expressions.....	103
Figure 38. An example for an annotated tweet	119
Figure 39. Feature engineering stages for sentiment analysis	127

Figure 40. The impact of phonological features on domain-problem	132
Figure 41. Rule-based classification approach	152
Figure 42. The performance of under-sampling techniques	155
Figure 43. The performance of seven fanatic-lexicons generated by PPMI method.....	157
Figure 44. The performance of seven fanatic-lexicons generated by the proposed- TFICF method.....	159
Figure 45. The impact of the proposed stylistic features	161
Figure 46. The impact of the proposed phonological features.....	163
Figure 47. Classifications using Unigrams Bag-Of-Stems Feature	165
Figure 48. Classification of bigrams bag-of-stems feature.....	166
Figure 49. Combining unigrams and bigrams bag-of-stems feature Classification.....	168
Figure 50. Combining unigrams and bigrams bag-of-emojis feature Classifications	169
Figure 51. Combining Punctuation, hashtag, emojis and stems features Classifications	171
Figure 52. The impact of the proposed syntactical features	172
Figure 53. Applying Bag-of-Words Syntactical features	174
Figure 54. Hybrid classification approach	175
Figure 55. Classifications using the proposed 16 features extracted from PPMI lexicons	177
Figure 56. Classifications using the proposed 16 features extracted from TFICF lexicons	179
Figure 57. Classifications using the proposed 32 features extracted from both PPMI and TFICF lexicons.....	180
Figure 58. Fanatic Classification Tool.....	187
Figure 59. Annotation Tool.....	188

LIST OF ABBREVIATIONS

API	:	Application Programming Interface
ASA	:	Arabic Sentiment Analysis
BPNN	:	Back-Propagation Neural Networks
CL	:	Computational Linguistics
CBOW	:	Continuous Bag-Of-Words
CNN	:	Convolutional Neural Network
CNNs	:	Convolutional Neural Networks
CFS	:	Correlation-Based Feature Selection
DT	:	Decision Tree
DNN	:	Deep Neural Network
GNB	:	Gaussian Naive Bayes
GA	:	Genetic Algorithm
hbc	:	Hosebird Client
IG	:	Information Gain
IDF	:	Inverse Document Frequency
KACND	:	King Abdul Aziz Center for National Dialogue
KNN	:	K-Nearest Neighbor
LDA	:	Latent Dirichlet Allocation
LLR	:	Logistic Linear Regression
LSTM	:	Long Short-Term Memory
LSTM	:	Long Short-Term Memory
MSA	:	Modern Standard Arabic
NB	:	Naive Bayes
NLP	:	Natural Language Processing
NEUNET	:	Neural Network
POS	:	Part of Speech
PMI	:	Pointwise Mutual Information
PPMI	:	Positive Pointwise Mutual Information
PCA	:	Principal Components Analysis
RNN	:	Recurrent Neural Network
SAFF	:	Saudi Arabian Football Federation
SER	:	Sentiment Emoji Ranking
SG	:	Skip-Gram
SFF	:	Sport Fanaticism Formalization
SFT	:	Sports-Fanaticism in Text
STD	:	Standard Deviation
SGD	:	Stochastic Gradient Descent
SVM	:	Support Vector Machine
SVM	:	Support Vector Machines
TF	:	Term Frequency
TF-IDF	:	Term Frequency- Inverse Document Frequency
TF-ICF	:	Term Frequency- Inverse Context Frequency
WSA	:	Word Sense Ambiguation

ABSTRACT

Full Name : [Mohammed Mugahed Yahya Al-Qmase]
Thesis Title : [Sentiment Analysis for Sports-fanaticism in Arabic Social Media Text]
Major Field : [Computer Science]
Date of Degree : [April 2019]

Sport fanaticism is one of the social problems that have negative impact on social fabric. It is a psychological emotion that carries a blindness hatred against the competitive (teams, players, etc.) combined with a blindness love toward favorite (teams, players, etc.) where the emotional attitudes overcome the mental attitudes. Social media sites (e.g. Twitter) have become common means for communication between sport fans. Therefore, understanding how social media text contributes to increase or decrease sports-fanaticism is essential. We need to formulate the concept of Sports-Fanaticism into suitable criteria, rules, attributes, aspects, features and indicators that allow applying Arabic sentiment analysis techniques to help in automatically detecting and measuring this phenomenon in social network sites. The target of this research work is to automatically classify Arabic texts under consideration into *fanatic* and *non-fanatic* emotion. This will help the interested researchers and specialists in detecting and measuring the degree of sport fanaticism automatically. After formulating the problem, the proposed formalism was used as a guide to build annotated corpora and fanaticism-lexicons. To build these resources, around 4 million tweets were collected. We have used Positive Pointwise Mutual Information (PPMI) and Term-Frequency Inverse-Context-Frequency (TFICF) methods to generated ten fanatic-lexicons where Our proposed TFICF method showed better performance than

PPMI in all experiments. The highest accuracy of TFICF is 91% and the highest accuracy of PPMI is 86%. Large-scale annotated corpora were also constructed. After building the required resources, we have conducted various experiments using different approaches, algorithms, and features. SVM, LR and NB classifiers were investigated. As one of the results, we have developed a classification tool that automatically evaluates a given text against sport fanaticism. such tool will help news sites to automatically filter messages posted by visitors to avoid sport fanaticism.

Keywords: Sport Prejudice, Sport Intolerance, Opinion Mining, Social Networks, Machine Learning, Arabic Processing, Text Mining.

ملخص الرسالة

الاسم الكامل: محمد مجاهد يحيى القماسي

عنوان الرسالة: تحليل مشاعر التعصب الرياضي في النصوص العربية في مواقع التواصل الاجتماعي

التخصص: علوم حاسوب

تاريخ الدرجة العلمية: أبريل 2019

التعصب الرياضي هو أحد المشكلات الاجتماعية التي لها آثار سلبية على النسيج الاجتماعي، فهو عاطفة نفسية محملة بالكراهية ضد الفريق المنافس مع التحيز الشديد للفريق الخاص؛ بحيث تغلب عاطفة الكراهية والتحيز على التفكير العقلاني. وبما أن مواقع التواصل الاجتماعي مثل تويتر (Twitter) أصبحت من الوسائل الأكثر شيوعاً للتواصل بين عشاق الرياضة؛ فقد ظهرت الحاجة لأداة تساعد في اكتشاف هذه الظاهرة في مواقع التواصل الاجتماعي وقياسها آلياً. نستخدم في هذا البحث تقنيات تحليل المشاعر لدراسة التعصب الرياضي في النصوص العربية المستخدمة في تويتر. وتهدف هذه الدراسة إلى تصنيف النصوص العربية آلياً إلى نصوص تتصف بالتعصب ونصوص لا تتصف بالتعصب. ويُساعد هذا التصنيف الباحثين والمتخصصين والمهتمين في اكتشاف وقياس درجة التعصب الرياضي في مواقع التواصل الاجتماعي آلياً. ولكي نستطيع بناء مصنف نصوص آلي للتعصب الرياضي، قمنا بصياغة مفهوم التعصب الرياضي في النصوص إلى معايير ومؤشرات ودلالات ملائمة تُسهّم في تعريف وتحديد ما هو النص الذي يحمل تعصباً، وما هو النص الذي لا يحمل تعصباً، وتمت صياغة المفهوم تحت إشراف متخصص في علم النفس الاجتماعي الرياضي. ومن ثم أُستخدمت هذه الدلالات كأساس لبناء الموارد المطلوبة لتطبيق تقنية تحليل المشاعر، مثل مكانز الكلمات (lexicons) ومكانز الجمل الموسومة (annotated corpora). ولأجل بناء هذه الموارد تم جمع ما يقارب من أربع مليون تغريدة من تويتر، وتم إعدادها لبناء مكانز جمل موسومة على شكل حزم بيانات موسومة (annotated datasets) تساعد الباحثين لإجراء بحوث في هذا المجال. وفي هذه الدراسة استخدمنا طريقتين لبناء مكانز الكلمات، الطريقة الأولى (Positive Pointwise Mutual Information) وهي الطريقة الأكثر شيوعاً لبناء مكانز الكلمات، بينما الطريقة الثانية هي طريقة تم

اقترحها في هذا البحث تسمى تكرار المصطلح-معكوس تكرار السياق (-Term-Frequency Inverse-Context-Frequency). وبعد مقارنة نتائج أداء الطريقتين تبين أن الطريقة المقترحة أفضل من الطريقة الأخرى حيث أن أعلى دقة أدى للطريقة المقترحة هو 91% بينما أعلى دقة أداء للطريق الأخرى 86%. وبعد أن تم إعداد الموارد المطلوبة لتطبيق تقنيات تحليل المشاعر، أجرينا عدة تجارب بعدة طرق وباستخدام ثلاث خوارزميات من خوارزميات الذكاء الاصطناعي وهي (SVM, LR and NB) لبناء نموذج تصنيف النصوص. وكننتيجة لهذا البحث قمنا بتطوير أداة تعمل على تصنيف النصوص العربية آلياً. ويمكن لعشاق الرياضة استخدام هذه الأداة لتجنب التعصب الرياضي أثناء كتابة الآراء. كما يمكن لهذه الأداة مساعدة المواقع الإخبارية الرياضية للتحقق من الرسائل والتعليقات التي ينشرها الزوار وحجب الرسائل التي تعاني من التعصب الرياضي.

الكلمات المفتاحية: التحيز الرياضي، التعصب الرياضي، التنقيب عن الآراء، الشبكات الاجتماعية، التعلم الآلي، المعالجة العربية.

CHAPTER 1

INTRODUCTION

“Sentiment analysis, also called opinion mining, is the field of study that analyzes people’s opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes” [1]. Research in sentiment analysis has an important impact on Natural Language Processing (NLP), management sciences, political science, economics, and social sciences where all are affected by people’s opinions [1]. There is an explosive growth of social media (e.g., reviews, forum discussions, blogs, microblogs, comments, and postings in social network sites) on the web. The contents in these media are increasingly used for decision-making by analyzing the posted data and extracting some knowledge for various proposes such as marketing [1]. Recently, sentiment analysis of Arabic has received extra attention from the research community [2]. The complexity of Arabic language and its linguistic features limit advancement in Arabic sentiment analysis [3]. Therefore, any advancement in this path will help. In this research, we use sentiment analysis to evaluate and classify Arabic texts in sport social media into fanatic sentiments and non-fanatic sentiments. Sport fanaticism is a social problem that has a negative impact on the social fabric [4]. Sport fanaticism is a psychological emotion with extreme hatred against the competing team and excessive love for the cheerful (own) team [5]. This emotion is controlled by sentiment not by mind [4]. Sport fanaticism is a negative trait, especially if

it becomes a social phenomenon [4], [5]. This research work aims to provide an automatic solution that helps in detecting and measuring this phenomenon in Arabic texts using sentiment analysis and machine learning techniques.

To handle this problem, we start by formalizing the concept of “*sport fanaticism in texts*” into criteria, definitions and indicators. Then, we build sentiment analysis resources using the proposed criteria. The required resources are labeled-corpora and fanatic-lexicons. Building resources requires gathering a collection of Arabic datasets from sport social media and annotating the datasets manually and systematically from the perspective of sport fanaticism. The opinions/ sentiments/ messages/ posts/ tweets are annotated as Fanatic (negative) or as Non-fanatic (positive) according to the proposed criteria and rules. After that, text preprocessing is needed to put social media posts in an appropriate format as the posts usually contain spelling errors, dialect, symbols, structure mistakes, etc. Therefore, we have used, developed, and modified preprocessing tools to do tokenization, cleaning, normalization, stemming, stop words removal, etc. In the next phase, different types of features are extracted and examined. In the classification phase, we examine some machine learning algorithms to do classification and build the model. Finally, we evaluate the model and implement a software prototype to classify text automatically.

1.1 Problem Statement

Sport fanaticism causes serious social issues that negatively affect the social fiber [4]. It can lead to negative relationships and society dismantling [5]. Social media and social communication sites increase sport fanaticism where these sites provide important means

for sport fans to communicate. Sport fanaticism in social media is an issue that requires more attention [6], [7].

1.2 Proposed Solution

This research work proposes a scientific model that automatically detects and classifies sport fanaticism in texts using sentiment analysis and machine learning techniques. The proposed model will automatically classify Arabic sports text into fanatic and non-fanatic text.

1.3 Research Goal

The goal template discussed by Wohlin et al. [8], has been used to describe our research goal. Our research goal is to **formalize** the concept of "*sport fanaticism in texts*" into criteria, rules and indicators **for the purpose of** building labeled-corpora, fanatic-lexicons, classification models and tools **with respect to** Arabic social media quality **from the point of view of** psychology and sociology **in the context of** sentiment analysis and machine learning.

The developed resources, models and tools will help to automatically classify Arabic sports text into fanatic or non-fanatic opinions/ sentiments/ messages/ tweets/ posts.

1.4 Research Objectives

In this section, we identify what we are trying to achieve. We have two main objectives:

- 1) Formalize the concept of “*sport fanaticism in text*” into practical definitions, criteria, rules and indicators of a “*sport fanaticism in text*” and validate the formalism by an expert in the field of sociology and psychology.
- 2) Design and develop models and tools that automatically classify Arabic sports text into fanatic or non-fanatic opinion/sentiment using machine learning and sentiment analysis techniques according to the developed formalism. To achieve these objectives, we need to
 1. Build Arabic resources (domain-specific: corpora, lexicons and dictionaries) according to the suggested definition and indicators of “*sport fanaticism in text*”.
 2. Choose the appropriate preprocessing level that achieves high performance.
 3. Extract and select the appropriate features that achieves high performance.
 4. Choose suitable classification approaches and algorithms that achieves high performance.
 5. Evaluate and compare the results.
 6. Build tool and web services for model execution.

1.5 Research Questions

This research work will answer the following significant questions:

1. How to formalize the concept of "*sport fanaticism in texts*" into criteria, rules and indicators for the purpose of building sentiment analysis resources and classification model?
2. How to classify Arabic text automatically into fanatic and non-fanatic opinions/sentiments/emotions according to the proposed definition and indicators?
3. What is the impact of over and under preprocessing on Arabic sentiment analysis?

1.6 Research Motivation

Some motivations that attract us to conduct our research in the field of Arabic sentiment analysis, Arabic processing and the “*sport fanaticism phenomenon in texts*” are:

1. The spread of sport fanaticism phenomenon in social media [5] [7].
2. The lack of resources of Arabic sentiment analysis [9] [10] [11] [3].
3. The lack of research in Arabic sentiment analysis [9] [10] [11] [3].
4. The Lack of domain-specific Arabic annotated corpus for sentiment analysis [11].
5. The Lack of domain-specific senti-lexicons for sentiment analysis in Arabic [10].
6. The Lack of preprocessing tools for Arabic sentiment analysis [11] [3].
7. The Lack of automatic tools for analyzing Arabic sentiment analysis in social media that help in decision-making.
8. As far as we know, there is no proposed method or model to detect sport fanaticism in text automatically (social media and news sites).

1.7 Research Scope

This research work is phenomenon-specific, domain-specific, language-specific, sources-specific, and time-specific. The following list specifies the attributes of the scope of this research.

1. *Phenomenon-specific*: The Phenomenon under consideration is “sport fanaticism in texts”.
2. *Domain-specific*: The addressed domain is Saudi football reviews, comments and tweets.
3. *Language-specific*: The targeted text language is modern standard Arabic and Arabic dialects.
4. *Sources-specific*: The used data sets are only reviews, comments, tweets, and/ or posts gathered from Twitter.com.

5. *Time-specific*: The gathered datasets were published in the period between (October 22, 2018) and (December 16, 2018).

1.8 Research Outcomes

The outcomes of this research work are annotated-corpora, fanatic-lexicons and classification models as describe below:

1. Two annotated corpora about sport fanaticism: The first corpus is the fanatic-corpus, which contains more than 250K tweets. This corpus was labeled automatically. The second corpus contains around 10K tweets and was labeled semi-automatically.
2. Seven annotated lexicons were developed. One lexicon was manually developed. It contains around 1.7K phrases. Three large-scale text lexicons were automatically extracted which are: text unigrams fanatic-lexicon, text bigrams fanatic-lexicon and text trigrams fanatic-lexicon. Three emojis lexicons were automatically constructed which are: emojis unigrams fanatic-lexicon, emojis bigrams fanatic-lexicon, and emojis trigrams fanatic-lexicon.
3. Classification models that help classify a given text to fanatic and non-fanatic text.

1.9 Research Challenges

Many challenges arise while working on Arabic sentiment analysis and Arabic natural language processing [9] [10] [11] [3]. Some of these challenges are parsing Arabic sentences, identifying entities and aspects, handling colloquial, lacking of preprocessing tools, handling misspelling and transliteration, handling negation, detecting opinion spam, resolving co-reference, handling reversed emoticons, handling Arabizi (e.g. hello Ali =

mar7aba 3ali), lacking of Arabic sentiment resources, handling comparative opinions, handling subjectivity classification, sarcasm detection, handling compound phrases and idioms, and handling implicit opinions.

1.10 Research Structure

The remaining parts of this thesis are structured as follows.

Chapter 2 (Literature Review): The purpose of the review is to provide definitions, context, and a clearer understanding of previous research in Arabic sentiment analysis and sport fanaticism.

Chapter 3 (Research Methodology): Abstract description about each stage in this work where it will help understanding what happened in each phase.

Chapter 4 (Sport-Fanaticism Formalization): Sport Fanaticism is defined and formulated into indicators/features where these indicators can be used as guide when building fanatic and sentiment resources.

Chapter 5 (Data Collection): Overview of the collected data is discussed. Statistical analysis on the collected data is described.

Chapter 6 (Data Preprocessing): The tasks of preprocessing on the data is discuss where cleaning, normalizing and stemming are applied. Over and under preprocessing is discussed.

Chapter 7 (Building Sentiment Resources): Describing the methods of constructing Fanatic-lexicons and annotated corpora.

Chapter 8 (Feature Engineering): Identifying and extracting five features-sets. The extracted features are described.

Chapter 9 (Polarity Classification): Applying three approaches: learning-based classification, Rule-based classification and hybrid classification. The classification results are discussed.

Chapter 10 (Dissection and Conclusion): The thesis results are discussed, and the research questions are answered. |

CHAPTER 2

LITERATURE REVIEW

To support the objectives of this research work, a literature review is conducted to provide fundamental background information about sport fanaticism and Arabic sentiment analysis. Background about sport fanaticism, related resources, feature engineering and classification is reviewed in this chapter.

2.1 Sport Fanaticism

Sports-fanaticism is one of the social problems that has a negative impact on the social fabric. Social media and social communication sites provide important means for sports fans to communicate [7]. Some works show that sports fanaticism in social media is an issue that requires more attention [12], [7]. Therefore, understanding how social text contributes to increase or decrease sport-fanaticism is essential. Alshehri et al. [13] summarized the definition of Castillo et al. [14] for fanaticism as an “extreme negative or positive **feelings** regarding the members of a particular group or social category”. Alshehri et al. [13] also stated the Young-Brueh’s [15] definition as “fanaticism consists of negative **attitudes** against a particular group or to any person who realizes that he belongs to this group”. Clearly, these definitions stated that fanaticism is attitude and feeling.

Many studies have examined the causes and effects of sport fanaticism. For example, King Abdul Aziz Center for National Dialogue (KACND) studied cause-and-effect of sports-fanaticism and the ways of dealing using dialogue [5]. Alshehri and Alrabaan [16]

examined the impact of some socialization systems on sports-fanaticism. Al-Tayyar [4] studied the sport fanaticism and its impact on family relationship. Ibrahim [17] studied the causes of public fanaticism associated with sports and suggested some solutions to the phenomena. Such studies and others confirmed that the sport fanaticism has a negative impact on social fabric. Other studies examined the impact of media and social media on sport fanaticism such as [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29]. Social media and social communication sites provide important means for sports fans to communicate [7]. Some studies showed that sport fanaticism in social media is an issue that requires more attention [12], [7]. The aims of some other research efforts were to find some indicators, standard levels, rules and measures to detect the sport fanaticism. Examples of these research efforts could be found in [30] [31] [32] [33]. As far as we know, there are no proposed indicators, rules and criteria that can be used to automatically detect “*Sport fanaticism in text*”. Therefore, we need to formalize the *sport fanaticism in text* into indicators that helps to automatically detect sport fanaticism in social network texts. Any tool, model, or technique that helps to automatically detect sport fanaticism in social media texts is appreciated and encouraged by government and society [5], [7].

The aim of this research work is to find indicators/rules/criteria of “sport fanaticism in texts” then, we will develop classification model using machine learning and sentiment analysis that helps to automatically detect sport fanaticism in texts.

2.2 Building Resources

This section discusses building sentiment lexicons and annotated corpora. It is emphasized in the literature that there is a lack in Arabic resources for domain-specific sentiment

analysis. Obviously, building Arabic resources for domain-specific sentiment analysis is valuable and appreciated. However, building sentiment lexicons, annotated corpora, ontologies, and other resources, is time and effort consuming. In this section, literature of building senti-lexicons and annotated corpora will be reviewed and discussed.

2.2.1 Building Sentiment Lexicons

The sentiment lexicon also called opinion-lexicon and sentiment-dictionary. It refers to a dictionary of words, phrase or idioms. These terms can be used to calculate the sentiment score for each document and then classify the documents to their high polarity. These words are considered as good indicators for sentiment analysis. They are called in the literature as *sentiment-words*, *opinion-words*, *polar-words* or *opinion-bearing words* [1]. There are common words that can be used to express positive or negative feeling, attitude and emotions. For example, *good*, *excellent* and *wonderful* can be used to express positive feeling where *bad* and *awful* is used to indicate negative feeling [1]. The lexicon can be used in different ways in sentiment analysis classification. For example, the sentiment score for each document is computed by summing up the lexicon scores of each term in the document considering negations and special cases. If the final score is greater than zero, then the document can be classified as positive, otherwise, it is negative.

There are four approaches to construct lexicons: manual-based, corpus-based, dictionary-based and translation-based.

For manually constructed lexicons, the sentiment lexicons are manually constructed and annotated by human. The annotation process is called polarity-determination or polarity-identification, where the annotator reads and understands several term-context information

and try to classify the term into positive or negative. Amiri et al. [34] have used a web interface to enable users to manually annotate terms using some guidelines and criteria. It might be more efficient to have crowdsourcing approach where more members contribute to decide the labels of the terms. Then, the agreement measure could be used to weight the terms. There are several issues with this approach. It is time and effort consuming. It is not usually used alone but it is combined with automated based approach [1]. This approach requires odd number of annotators greater than two to measure the agreement. Some terms can be used as positive in one domain, but negative in other domain [1]. This approach is usually used to construct and prepare seed senti-words to extract other words automatically.

For dictionary-based approach, dictionaries with synonyms and antonyms are used to construct and compile opinion-words. First, seed of polar words with known positive and negative sentiments are collected manually or semi-automatically. Then, the seed words are used to extract other words using bootstrapping approach. They are searched in the dictionary for synonyms and antonyms to cumulative and grow up the seeds. After the iterative process is completed, the manual inspection is used to clean and validate the collected list. Several researchers have used this approach, some of which are Aldayel et al. [35] and Mahyoub et al. [36].

In cross-language based approach or translation-based approach, an already constructed lexicon for a specific language is translated to another language. There are many attempts to translate high quality English senti-lexicons to Arabic and other languages (See Al-Moslmi et al. [10] and Mohammad et al. [37]). This approach is simple and depends on the quality of the translation and the quality of the original lexicon.

For corpus-based approach, a corpus is used to detect new words through a rule-based method. The seed sentiment-words are used with some conjunction-words such as (*and*, *but* and *however*). For example, each seed word can detect other words with the similar orientation if they have “*and*” conjunction between them or opposite orientation if they have “*but*” conjunction. This helps to cumulate and grow up the seed words using corpora. There is another method which is called a statistical-based method. This method uses statistical association measures to compile new sentiment-words using a large enough labeled-corpora. Any word can have a high positive score if it appears in many positive documents. On the other hand, a word can have a high negative score if it appears in many negative documents. This method assumes that a document can be labeled as a positive document if it has many positive words, and it can be labeled as a negative document if it has many negative words. Under this assumption, some works use *Pointwise mutual information* (PMI) to compute the polarity-score. The PMI can be used to automatically generate lexicons where a sentiment score of each word can be computed by mutual information.

Figure 1 summarizes these four approaches in constructing senti-Lexicons.

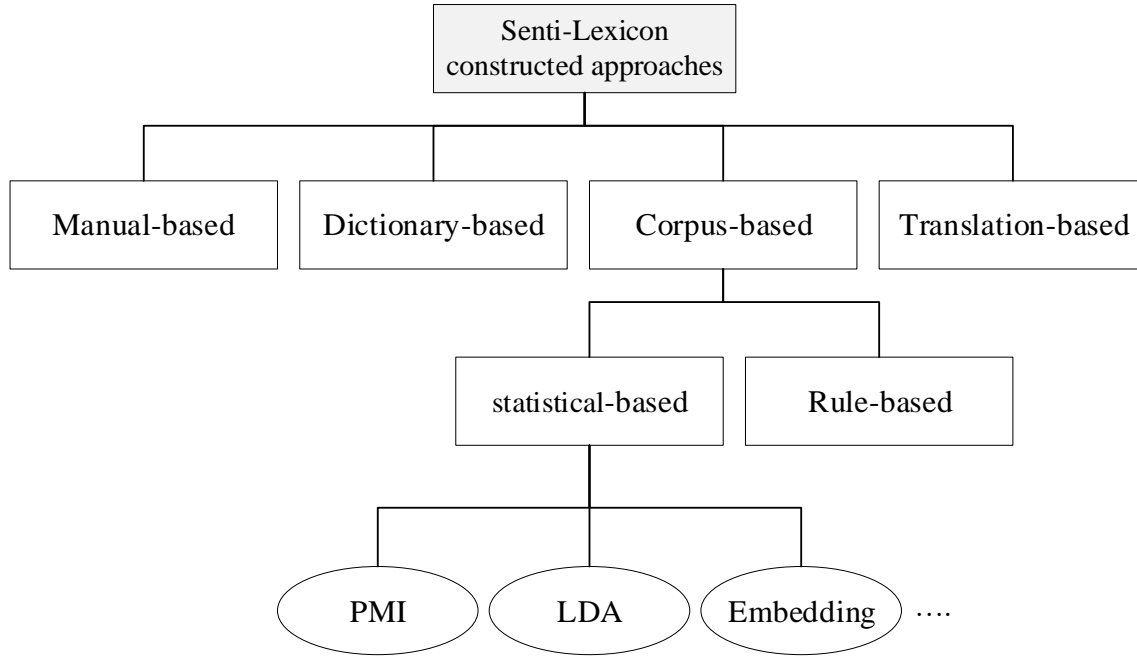


Figure 1. Senti-lexicon constructed approaches

The following presents some efforts to build sentiment lexicons.

In [38], El-Beltagy and Ali have manually collected 380 seed words. Then, they have used a corpus-based approach to extend the seed words and automatically construct general-domain senti-lexicon. To extend the seed-words, they have used Twitter APIs to extract 1000 tweets for each entry in the seed words. As they have mentioned, they collected 2700000 tweets. After that, they have used conjugated patterns to detect new words. Twitter APIs were utilized to search for each seed word (w). If the collected tweet having word w with the following patterns: (w و c) or (c و w) or ($يا$ w $يا$ c) or ($يا$ c $يا$ w), then the new word c can be considered as a candidate sentiment term with the same polarity as w . For example, the patterns ($يا$ w $يا$ c) have detect the following phrase ("يا مؤدب يا محترم") where we can notice that the word w and the word c have the same orientation. This process helped to extend the seed words to 4,392 senti-word which means that one word in one

orientation can detect many words in the same orientation. In addition, two corpus-based algorithms to weight and score each word were proposed. Two annotated datasets were used to evaluate the performance of the scored lexicon. The highest accuracy obtained was 83.8%.

ElSahar and El-Beltagy [39] have proposed extraction patterns to extract dialect subjective terms for sentiment analysis. They have applied corpus-based approach to extract senti-words. To build the patterns, they have used the entities: negator, demonstrative pronoun, intensifier, person reference, personal pronoun, conjunction and strong subjective. The proposed patterns were used to extract subjective terms. After building the patterns/rules, the proposed patterns were applied on the corpus with 7.5M Egyptian tweets. The applied patterns extracted around 633 unique terms. Then, three annotators classify the extracted terms into positive, negative and not-a-sentiment. This process had resulted in a total of 377 labeled correctly terms. To validate the annotated terms, they have used around 3M tweets labeled automatically using 2K seed words collected from prebuild lexicon [40] [41] [38]. The tweets that have mix positive and negative seed words are excluded. Then, PMI was applied on the tweets to measure the co-occurrence of the extracted terms using patterns in the positive and negative tweets. If difference between the PMI value of a term in the positive tweets, and the PMI value of the same term in the negative tweets is less than a specific threshold α , then the term is ignored. Their results have showed that 344 out of 377 terms was validated to be correctly labeled.

In [42], ElSahar and El-Beltagy have used a corpus-based approach to extract 2k mult-domain senti-terms. Firstly, a dataset with approximately 33K was scrapped form multiple domain websites. The user rate of each review was used to label the dataset. Feature

selection technique was applied using 1-norm *support vector machines* (SVM) to select a most significant feature. All unigrams and bigrams terms were used as features. The L1 regularization results in sparse weight vectors with 6708 selected terms. After that, the terms were manually filtered to end up with 1913 terms. Then, they have evaluated the terms using different classification algorithms. The average accuracy of 2-class classification was 78.2% while the average accuracy of 3-class classification was 55.4%.

El-Beltagy [43] has generated corpus-based lexicon with 5953 senti-words and senti-phrases, called NilULex. The generated lexicon was developed gradually and presented in [38] and [39]. It has 563 negative phrases, 416 positive phrases, 3693 negative words, and 1281 positive words. Many entities of the lexicon were collected manually. Two annotated datasets of tweets were used to evaluate the proposed lexicon. One of them was the Saudi dialect dataset and the other one was the Egyptian dialect dataset [42]. In the evaluation, the proposed lexicon was compared to another lexicon called EmoLex [44]. The highest classification results were obtained when the lexicon was applied on Saudi dialect dataset with 2-class classifications. The highest results showed that the proposed lexicon outperformed EmoLex with 89.4% accuracy on cross validation, while combined both lexicons obtain highest value on test dataset with 82.3% accuracy. The same author, El-Beltagy [45], has proposed an approach to automatically score each entry in this lexicon using corpus. The scored lexicon, *WeightdNileULex*, was generated. A Twitter corpus was utilized to compute the co-occurrences between terms and determine the score of each term. The corpus that was used contains 5000 tweets. The results showed that the scored forms outperformed unscored lexicon.

Youssef and El-Beltagy [46] have used word embedding to automatically generate Arabic senti-lexicon. They have utilized the corpus-based lexicon NileULex [43] as seed words and have used word embedding to expand them. The idea was to utilize the word embedding to generate related words to the seeds (e.g. Synonyms, antonyms). The embedding words helps to bring the words that can appear in the same context to gather by assigning high similarity value to them. Similar terms can be presented closed to each other in the vector space. Each term in the seeds was used as search token to find new similar words in the embedding. Then, the candidate words generated from the embedding were automatically filtered. A confidence value was selected to decide which term should be added and which term should not be added to the lexicon. The confidence value was calculated to reflect how much a candidate word is related to a seed word. If the ratio is greater than a specific threshold, then, the generated words will be added to the lexicon. To determine the polarity, the ratio of how many candidate words appeared in positive and the negative terms in the seed words, was computed.

Al-Moslmi et al. [10] have utilized MPQA senti-lexicon [47] to generate seed words. MPQA is an English senti-lexicon where it was translated to Arabic (MSA) using Google [48] translation. Two review corpora were utilized to extract dialectical senti-terms. They have manually extracted a list of opinion-terms from those two Arabic review corpora. Then, they have utilized SentiWordNet [49] to extract some synonyms. Finally, a list of 3880 positive and negative polarities with 13,760 synonyms were collected. Two scoring methods were followed: manual scoring and automatic scoring using a corpus. For manual scoring, three Arabic native annotators have contributed to score each term with value between -5 (extremely negative) and $+5$ (extremely positive). Then, annotators

communicated to resolve conflicts. For automatic scoring, two review corpora were utilized. The score was obtained by analyzing the occurrences of each word in the positive and negative reviews. They used the frequency ratio of each word in the positive and negative reviews to obtain the score.

Mohammad et al. have constructed eight senti-lexicons [37]. Three of the eight were corpus-based lexicons and the other five were translation-based lexicons. The corpus-based lexicons were constructed from around 1 million tweets where PMI was used as statistical association measure to calculate the scores of the entities in the lexicons. To calculate the score, annotating the collected tweets was required. For the first lexicon, Arabic-Emoticon-Lexicon, the tweets were annotated using two emoticons. For the second one, Arabic-Hashtag-Lexicon, the tweets were annotated using 77 hashtags keywords. For the third lexicon, Arabic-Hashtag-Lexicon (dialectal), the tweets were annotated by dialectal seed words collected by Refaee and Rieser [50]. On the other hand, the translated-based lexicons were constructed by translating the following English lexicons: AFINN [51], Bing Liu Lexicon [52], MPQA Subjectivity Lexicon [47], NRC Emotion Lexicon [53] [44], NRC Emoticon Lexicon (Sentiment140 Lexicon) [54] [55], and NRC Hashtag Sentiment Lexicon [54] [55]. Then, constructed lexicons were tested. The highest performance reached 65.3 % accuracy on the Arabic Hashtag Lexicon (dialectal).

Mataoui et al. [56] have built three lexicons for Algerian Dialect. They have used a translation-based method to construct the lexicons. For the first lexicon, they have used the Egyptian lexicon of a research group in Nile University. They have first removed all the words that cannot be translated to Algerian. Then, the selected words were translated from Egyptian to Algerian. They have ended up with 2380 positive terms and 713 negative

expressions. They have used MSA dictionary to construct the remaining two lexicons, which were negation and intensification words. Then, they have built rule-based classifier that utilizes the three lexicons to classify the text. For testing, 7698 Facebook comments were collected and manually annotated. They have used the annotated corpus in testing. They have reached 79.13% accuracy.

Aldayel and Azmi [35] have used a dictionary-based and a manual-based method to extract 1500 senti-words with 1000 negative terms and 500 positives terms. They utilized SentiWordNet [49] to collect the senti-words. They have used the extracted senti-words as seed words to automatically annotate the collected tweets. The purpose of the collected polar-terms was to automatically annotate the senti-corpus. For classification, a bag-of-words with ngrams feature-representation was extracted. The TFIDF was used as weighting schema for each ngrams feature. Then, SVM was applied to build 2-class classification model. The highest accuracy obtained was 84.01%.

Al-Twairesh et al. [57] have proposed two lexicons, *AraSenti-Trans* using translation-based approach, and *AraSenti-PMI* using corpus-based approach. The translated lexicon contained 59,525 positive terms and 71,817 negative terms while the corpus-based lexicon contained 56,938 positive terms and 37,023 negative terms. For the used translation-based method, two English lexicons, Liu [52] and MPQA [47], were utilized. To construct the *AraSenti-Trans* lexicon, 7,135,331 tweets were collected using polar-keywords and polar-emoticons. Then, the MADAMIRA tool [58] was utilized to generate stems and English glosses. The English glosses were compared to the English words exist in the English lexicons. If all glosses words of the target word have positive polarity in both lexicons, then the word considered as a positive word. On the other hand, if all glasses words of the

target word have negative polarity in both lexicons, then the target word considered as a negative word. Neutral can be determined if the above two cases failed. For corpus-based approach, PMI was utilized. PMI is a statistical association measure where the association between any term and the positive and the negative categories is calculated as follow: $\text{SenScore}(w) = \text{PMI}(w, \text{pos}) - \text{PMI}(w, \text{neg})$ where w is a term in the lexicon. The best F1-score that was obtained by PMI lexicon with 89.58%.

Mohammad et al have examined [59] the impact of translation on sentiment analysis. They have applied two-ways examination. First, they have examined the performance of Arabic language text translated to English and have applied an English sentiment analysis system. Second one, they have examined the performance of translating annotated-corpora and lexicons from English to Arabic and have applied Arabic sentiment analysis system. For English translated to Arabic, 1200 Levantine dialectal sentences from the BBN Arabic-Dialect-English Parallel Text [60] were selected. For Arabic translated to English, 2000 Syria tweets were collected and annotated. Also, three Arabic lexicons and 5 English lexicons were created. In the first experiment, they have tested the performance of Arabic sentiment analysis systems using the following resources: the Arabic training corpus, the Arabic sentiment lexicon, and the Arabic translation of the English lexicon. In the second experiment, they have applied state-of-the-art English sentiment analysis systems of Arabic resources that were translated to English. They have found that the English resources translated to Arabic help improve Arabic sentiment analysis systems.

Alshahrani and Fong [61] have adapted a topic modeling technique called *Latent Dirichlet Allocation* (LDA) to construct a domain-specific senti-lexicon on Saudi stock market. LDA is an unsupervised learning algorithm that can be used as a generative statistical model to

classify text into unobserved categories. Such technique is being used in topic-modeling to classify given documents into different topics. It statistically calculates the associations between terms, terms-document and terms-topic. In addition, it statistically computes the relation between words and groups them into several topics. A senti-corpus that has been manually labeled, was used. Then, all positive documents were aggregated into one positive document, and all negative documents were aggregated into one negative document. Two-documents were the result of aggregation: one contained all positive documents and one have all negative documents. The big two documents were converted into document-term matrix which was used as input to LDA. The aggregation step helps LDA to do well in polarity detection where it assumes that each document is related to one topic. They set 2-topic as a parameter to LDA to inform the algorithm to group the terms into two groups. The output of LDA was two set of words with their probability, one for each orientation. The authors have pointed out that the result of this approach was promising.

Ihnaini and Mahmuddin have proposed a bootstrapping approach to expand the existing lexicon automatically from labeled and unlabeled tweets [62]. They have integrated six published senti-corpora into one. The aggregated corpus was split into training and testing. They have constructed a basic-lexicon with 4266 seed senti-words from three English lexicons. Then, they have proposed some rules to automatically extract new senti-words to expend the basic lexicon from training dataset. They have compared 10 Arabic lexicons on their method to test their approach. The results showed that the expended lexicon improved the accuracy.

Table 1 summarizes the discussed sentit-lexicons approaches.

Table 1. Comparing Sentiment Lexicons

Ref.	Lexicon constructed methods				Language	Domain	Size	Available Access Link
	Manual-based	Dictionary-based	Corpus-based	Translation-based				
[45]	√	-	√	-	MSA + Egyptian + others	general	5953	[63]
[46]	√	-	√	-	MSA + Egyptian + others	general	36775	NA
[61]	-	-	√	-	MSA + Saudi	Saudi stock market	10000	NA
[62]	-	-	√	√	MSA + Dialect	general	4,266	NA
[64]	√	√	-	-	MSA + Saudi	general	14,000	NA
[10]	√	√	√	√	MSA + Dialect	general	3880 + 13,760 synset	[65]
[37]	-	-	√	√	MSA + Dialect	general	Three: 43,304; 21,964; 20,128	[66]
[56]	-	-	-	√	MSA + Algerian Dialect	general	3093	NA
[59]	-	-	√	√	MSA + Dialect	general	Three: 43,304; 21,964; 20,128	[66]

Ref.	Lexicon constructed methods				Language	Domain	Size	Available Access Link
	Manual-based	Dictionary-based	Corpus-based	Translation-based				
[43]	√	-	√	-	MSA + Egyptian	general	5953	[63]
[57]	-	√	√	√	MSA+ Saudi Dialect	general	225329	[67]
[35]	√	√	-	-	MSA +Saudi	general	1500	NA
[42]	-	-	√	-	MSA + Dialect	general	1913	[68]
[36]	-	√	-	-	MSA	general	7576	NA
[39]	-	-	√	-	MSA + Egyptian	general	344	[63]
[69]	-	√	-	-	MSA	general	28,780	[70]
[38]	√	-	√	-	MSA + Egyptian	general	4,392	[40]
[71]	√	√	-	-	MSA	newswire domain	29,816	NA
[50]	√	-	-	√	MSA + Dialect	general	4,422	By Email

From the previous summary, we can observe that the manual-based lexicons have required time and effort to construct them. The lexicons, built with manual-based method, have small size compare to others. This approach is not usually used alone. From the above studies, that manual-based method can help to prepare seeds of senti-words where these

seed-words can be extended using other methods. Manual-based method can provide good quality senti-words specially when many native and expert annotators contribute to build the lexicon.

We noticed that the dictionary-based approach provides a good coverage. It can help to extend the lexicon extracted using manual-based approach. Most of the used Arabic wordNets are translated from English to modern standard Arabic (MSA), and do not contain dialects. Therefore, MSA seed-words were needed in order to bootstrap such WordNet to obtain synonyms and antonyms. Although the dictionary-based approach is simple and straightforward, it seems that the result of this approach can generate only MSA senti-lexicons. In addition, it requires postprocessing to clean and validate the generated lexicons. This method can have another limitation where the extracted lexicon can be used for general-domain sentiment analysis. It might be a good idea to use a domain-specific/ domain-oriented WordNet to overcome this limitation. This idea depends on the availability of domain-specific WordNet.

We can observe that translation-based approach is the simplest method as it only requires choosing a high-quality translation and high-quality source-lexicons. When automatic translation is used, this method provides a good coverage compared to manual-based approach. However, it seems that the result of this method might not have dialectal senti-terms because most of translators translate only the source-language lexicon to MSA Lexicon. The future of this approach is promising when Arabic colloquial translators are available. From the literature, Mataoui et al. had translated Egyptian-lexicon, called NileULex, to Algerian-lexicon [56]. However, this translation was achieved manually and

has the limitation of manual-based approach. In dictionary-based approach, translation-based approach requires postprocessing to clean and validate the generated lexicons.

The main limitation of Dictionary-based and translation-based is that the generated lexicons can be only MSA. One idea, that can be used to overcome this limitation, is to utilize the available MSA-to-colloquial convertor where the extracted lexicon can be extended by adding the automatically converted terms. There are some efforts to build MSA-to-colloquial convertor [72] such as the work of Mahgouba, and Shaabanb [73]. As far as we know, there are no efforts to overcome this limitation to generate dialectal senti-lexicon.

The corpus-based approach can also help to overcome the above limitations where the corpus is a source to extract senti-words. To generate domain-specific lexicon, domain-specific corpora are needed. In addition, we can generate rich-dialectal senti-lexicon from corpus that contains informal and slang terms. Not just this, we can also generate region-specific dialectal senti-lexicon using region-specific corpora. The quality of this approach depends on the quality of the corpus used to generate the senti-lexicon. This method requires a good and big enough corpus.

The corpus-based approach has two sub-methods which are statistical-based method and rule-based method. The statistical-based method requires labeled corpora and assumes that the positive reviews/documents have positive senti-terms, and the negative reviews/documents have negative senti-terms. This method uses statistical association measures to compute the weight of senti-words and classify them. The advantage of this method compared to rule-based method is that It can provide statistical score of each senti-

word. On the other hand, rule-based method does not require labeled corpora. It uses some conjunctions such as and/but to find synonyms and antonyms [38] [9].

We can deduce that the statistical-based methods can provide good results when the size of the document is low, and the size of the used corpus is big-enough to infer the polarity and the score. This depends on the assumption that positive reviews/documents have positive senti-terms, and negative reviews/documents have negative senti-terms. As a result, we can deduce that a small document has high probability to have one opinion and one polarity. Therefore, the sentence-level labeled corpus and the tweet corpus can provide a good source to generate high quality senti-lexicon using statistical-based methods.

We can observe from the literature that there is a lack of domain-specific senti-lexicons. Most of available lexicons were built for general-purpose sentiment analysis. Therefore, Research community would appreciate any domain-specific lexicon. According to Liu, The senti-words can have opposite orientation in different domains [1]. Even inside the subdomains, for example, the words “sucks” can imply a positive sentiment in one domain e.g. “*This camera **sucks***” and can indicate negative sentiments e.g., “*This vacuum cleaner really **sucks***” [1].

Although the number of available lexicons is increasing, they are not sufficient to create more stable NLP resources such as Arabic WordNet. Therefore, more senti-lexicons can have positive impact on some NLP application and Arabic processing.

In this section, we have discussed senti-lexicon constructing methods. we can categorize these methods into four: manual-based, dictionary-based, translation-based and corpus-based. One observation is that the manual-based provides a good quality, but low coverage.

The high coverage senti-lexicon can be achieved automatically using other methods. Although the dictionary-based and the translation-based are simple and straightforward approaches to generate high coverage senti-words, these approaches have limitations to generate dialectal senti-lexicons. On the other hand, the corpus-based approach could be used to overcome this limitation. However, this approach requires high-quality and big-enough corpora. In addition, when statistical-based methods are applied, a labeled-corpus to generate the lexicons is required. In the future work, we plan to examine some of the ideas that have been suggested in this section to build sentiment lexicons. Moreover, we will extend the literature to cover more techniques used to generate Arabic senti-lexicons.

2.2.2 Building Annotated Corpora

An annotated corpus in sentiment analysis is a collection of comments, reviews, posts, tweets, sentences or any type of opinions' representations where each entity is associated with its orientation or sentiment score. Each entity is labeled with one of two labels (positive or negative), three labels (positive, negative or neutral), or more. Building annotated corpora for Arabic sentiment analysis require three tasks: collecting the needed data, preprocessing the collected data, and labeling each entity. In this section, methods of building annotated corpora will be reviewed and discussed. Data can be collected from many sources such as Facebook, Twitter, YouTube, etc. Some data sources provide APIs to access and retrieve data such as Twitter streaming APIs. After retrieving data, a preprocessing task is required to clean and remove duplicate and irrelevant entities and objects. Labeling collected data can be done in three approaches: manually, semi-automatically and automatically. Manual annotations for Arabic corpora require native Arabic speakers to read the entities and label them. The process of annotation (labeling)

can be done by a single annotator as in the research work of Mountassir et al. [74], or by more than one annotator as the case in the research work of Elouardighi et al. [75]. However, more annotators usually provide labels that are more reliable where each entity is labeled under crowdsourcing agreement. It is better to have odd annotators to break the conflict. While there is a lack of Arabic resources, an added annotated corpus, even with one annotator, can help. Although the output of manual annotation provides useful resources, the resources generated manually are usually limited to small size corpora compared with the resources generated by automatic approaches as can be noticed by reviewing Table 2. Moreover, Manual annotation is an effort and time-consuming task. The following literature summary gives an overview of used methods to construct annotated corpora.

Mountassir et al. have built two corpora and manually annotated them by a single annotator [74]. The first corpus contained 611 documents about Arab support for the Palestinian affair. The second corpus included 468 documents about a famous historical movie. The collected dataset was mixed content of Modern Standard Arabic text and Dialect text.

Abdellaoui and Zrigui have automatically collected a corpus using emojis [76]. First, they have used emjoitracker [77] to select 20 top-most emojis. Then, they have used Sentiment Emoji Ranking (SER) [78] to choose 10 most subjective emojis from the 20 emojis. After that, they have configured Twitter streaming API to collect tweets by filtering them using the 10 selected emojis. After preprocessing data, they have utilized Ar-SeLn [69] sentiment lexicon to label the collected dataset. The annotation was done by accumulating the polarity of positive and negative tokens. Ar-SeLn was used to count number of positive and negative tokens in each tweet. The label of each tweet was determined by the highest count.

As a result, 3,122,615 tweets were annotated as positive tweets, 2,115,325 tweets were annotated as negative tweets and 378,003 tweets were labeled as neutral. To validate the annotation, they have randomly extracted 1000 tweets from each class and relabeled them manually by 2 native Arabic speakers. The highest error rate that has been found was in neutral set with 11.3%. Then, the proposed corpus was compared with ASTD [79] corpus. Both datasets were split into 70% training and 30% testing and examined using several machine-learning algorithms. The constructed corpus showed better performance.

Elouardighi et al [75] have collected 10254 Facebook Political documents about Moroccan's legislatives elections using Facebook Graph API. The collected data were mixed contents of Modern Standard Arabic and Moroccan Dialectal Arabic. Through preprocessing, the collected dataset was cleaned and normalized. Ligh10 stemmer was applied to reduce the number of features. The comments were annotated manually using crowdsourcing where 6581 comments were annotated as negative and 3673 were annotated as positive. The constructed corpus was tested by applying three classification algorithms: decision tree, SVM and RF. These algorithms were applied on ngrams features with TFIDF weighting-schema. The highest F1-score achieved was 86% by SVM with unigrams and bigrams using TF-IDF weighting schema.

ElSahar and El-Beltagy have automatically annotated, five corpora [42]. User rating on the reviews was extracted and normalized into positive, negative, or mixed. Each review was rated on a scale of 1 to 5 stars where the authors have mapped the scale into a sentiment polarity. The collected data was about movies, restaurants, hotels, books and products. The developed corpora contained around 33K reviews entities. Different combination of features, classifiers and dataset setups were explored which lead to 3K experiments. The

experiments aimed to evaluate the usefulness of the developed resources. According to error analysis, they have found that the document length affects the accuracy.

Al-Twairesh et al. have conducted a research to prepare a corpus for Saudi tweets [80]. Using some emoticons and keywords, they have collected 2M tweets from Twitter. They have selected 17,573 Saudi tweets from the 2M corpus. Then, they have manually annotated the selected 17,573 tweets by three Arabic native speakers. One of four labels (positive, negative, neutral or mixed) was used to label each tweet. The results of annotation were validated by Kappa statistics measure which is used to measure the reliability of the annotation. Support Vector Machine (SVM) classifier was used to evaluate applicability of the developed corpus in sentiment analysis. Several experiments were conducted where 2-class, 3-class and 4-class classification were examined. The best result was 62.27% accuracy with 2-class classification and term-presence feature schema. However, the result of classification was not high. The reason for that might be the data was imbalanced and SVM is highly sensitive to imbalanced data [74].

Abdulla et al. have constructed an annotated corpus of 2000 tweets [81]. The corpus included 1000 positive tweets and negative 1000 tweets. The collecting method was through twitter crawling. After that, two human experts have annotated the collected tweets. When a disagreement between the two annotators has appeared, they have consulted a third expert.

Refaee and Rieser have used Twitter Streaming APIs to collect tweets for their corpus [50]. The authors have applied preprocessing tasks for cleaning and removing duplicates. The cleaned corpus was manually annotated by two Arabic native speakers. 6,894 tweets were

labeled into four labels: 833 positive, 1,848 negative, 3,685 neutrals and 528 mixed. The results of the annotation were validated by Cohen's Kappa to measure the degree of agreement among annotators. The Kappa score reached $\kappa = 0.816$, which indicated reliable annotations.

Refaee has used emoticons and hashtags with pre-determine polarity to collect and automatically label tweets [82]. The author has collected and annotated 2.5M tweets. The annotation process has resulted in 55,076 neutral, 660,393 positive, and 457,963 negative for emoticon-based (emo), and 55,076 neutral, 59,990 positive, and 70,170 negative for hashtag-based (hash).

Al-Moslmi et al. have constructed an Arabic multi-domain sentiment corpus (MAC) [10]. The corpus included 8860 reviews annotated as positive and negative by two Arabic native speakers. The corpus was collected from different sources where it contained mixed-contents MSA and dialect. To validate the corpus, the researchers have applied five classifiers to examine its usefulness on sentiment analysis.

Table 2 summarizes the above reviewed corpora and annotation methods.

Table 2. Comparing Annotated Corpora

corpus	size	Source	Classes	language		domain	Annotation Method		Annotation mean	Available
				MSA	Dialect		Manually	Automatically		
[74]	611	Aljazeera	2	√	Multi	Palestinian affair	√	-	1 annotator	No
[74]	468	Aljazeera	2	√		historical movie	√	-	1 annotator	No
[76]	6M	Twitter	3	√		general	-	√	lexicon-baed	Yes
[75]	10K	Facebook	2	√	Moroccan	Political	√	-	crowdsourcing ⁶	Yes
[42]	15K	TripAdvisor	3	√		Hotel	-	√	User Rating	Yes
[42]	11K	Qaym	3	√		Restaurant	-	√	User Rating	Yes
[42]	1500	Elcinema	3	√	Multi	Movie	-	√	User Rating	Yes

corpus	size	Source	Classes	language		domain	Annotation Method		Annotation mean	Available
				MSA	Dialect		Manually	Automatically		
[42]	15K	Souq	3	√	Multi	Product	-	√	User Rating	Yes
[42]	63K	GoodReads	3	√	Multi	Book	-	√	User Rating	Yes
[80]	17K	Twitter	4	√	Saudi	general	√	-	3 annotators	No
[81]	2K	Twitter	2	√	Jordania	general	√	-	3 annotators	Yes
[50]	6,894	Twitter	4	√	Multi	general	√	-	2 annotators	Yes
[82]	250 K	Twitter	3	√	Multi	general	-	√	Keywords and emoticons	Yes

corpus	size	Source	Classes	language		domain	Annotation Method		Annotation mean	Available
				MSA	Dialect		Manually	Automatically		
[10]	8860	Jeeran Qaym Google Play Twitter Facebook	2	√	Multi	Art and Culture Bakeries and Goodies Cafes Fashion Financial Services Hotels Restaurants Shopping Travel and Tourism Doctors Education Entertainment Health Care Software applications Political	√	-	2 annotators	Yes
[83] [59]	2K	Twitter	3	√	Levantine	general	√	-	crowdsourcing	Yes
[43]	6K	Twitter	3	√	Egyptian	general	√	-	3 annotators	by request

corpus	size	Source	Classes	language		domain	Annotation Method		Annotation mean	Available
				MSA	Dialect		Manually	Automatically		
[84]	500	Arabic blog sites	2	√	-	movies	-	√	User Rating	Yes

2.3 Feature Engineering

In machine learning, data is analyzed by machine learning algorithms to generate a model. The model in machine learning refers to knowledge extracted during the learning process. In the learning process, mathematical models are fitted on data to derive domain-knowledge or make predictions [85]. In order to fit such models on some given data, the data should first be converted into suitable and useful features. The process of converting data to useful features is called feature-engineering [85]. Feature engineering could be understood by explaining both terms: “feature” and “engineering”.

Features refer to suitable representations of data. Zheng et al has defined a feature as “*a numeric representation of an aspect of raw data*” [85]. It is also called *input variable* or *attribute* [86]. Specifically, it indicates to a meaningful attribute to the domain-problem. In other words, a feature could be any attribute that is useful for the model. It describes some aspect of individual data objects [87]. In the Oberman’s book “a feature is an individual measurable property or characteristic of a phenomenon being observed” [88].

In the other hand, engineering is “the process of utilizing knowledge and principles to design, build, and analyze objects” [89]. Engineering is “the application of scientific knowledge to solve problems in the real world. While science (physics, chemistry, biology, etc.) allows us to gain an understanding of the World and the Universe, Engineering enables this understanding to come to life through problem solving, designing and building things” [90]

From the definition of the two words “feature” and “engineering”, we can deduce the definition of feature engineering as a process of analyzing the data and designing what the model’s input X should be. In other words, it is a process of turning the raw data into useful and suitable input X to model. The following paragraph represent some of state-of-art definitions of feature engineering:

Zheng et al. stated that “feature engineering is the act of extracting features from raw data and transforming them into formats that are suitable for the machine learning model” [85].

In other words, Brownlee has defined feature engineering as “the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data” [91]. Similarly, Zaidi has defined feature engineering as “the process of using domain knowledge of the data to create features that make machine learning algorithms work” [92] [93]. Zheng and Casari have defined feature engineering as “a process of formulating the most appropriate features given the data, the model, and the task” [85].

Feature engineering is a vital stage in the machine learning pipeline [85]. It has high impact on the performance of the predictive models. However, it requires good experience on

domain-knowledge and data analytics [94]. Feature engineering stage usually includes set of tasks and processes such as features identification, feature generation, feature extraction, feature transformation, feature selection and feature evaluation [87]. The process of feature engineering can be done manually or automatically [94]. The works on automatic feature engineering is concentrated on providing a generic approach for automatically extracting a large set of features and selecting an effective subset of the extracted features [87]. The following summary gives an overview of applying different type of features in the literature.

Al-Twairesh et al. have examined the impact of different types of proposed feature sets [95]. Initially, they have identified semantic, syntactic and stylistic feature sets. For semantic features, they have extracted them using Arabic senti-lexicon called AraSenTi lexicon [57]. For syntactic and stylistic feature sets, they have extracted them from the Saudi tweet corpus that has been used for training and testing. The corpus contained 17573 tweets that were annotated manually into four labels: positive, negative, neutral and mix. Then, feature selection techniques were applied on sets of extracted features. The selection process was started using all features and was reduced gradually. The impact of the feature selection on 2-class, 3-class and 4-class were examined. SVMs with linear kernel were applied. They have examined the change of the performance when each feature is removed. On a 2-class classification, they have found the following features harm the performance: tweetLength, hasPositiveEmoticon, hasNegativeEmoticon, hasNegation, hasIntensifier, and hasDiminisher. On 3-class and 4-class classification, they have noticed that the performance was improved when removing: hasPositiveEmoticon and hasNegativeEmotion.

Refaee has examined the impact of six sets of features which were: syntactic, morphological, semantic, affective-cues, language-style and twitter-specific [82]. For semantic features, three lexicons were emerged which were ArabSenti [96], a translated version of MPQA [47] and in-house Twitter-based lexicon. To extract social-signals and affective-cues features, six manually created dictionaries were used which were: laughs, dazzle, prayer, regret, sigh, and consent. After designing and extracting the features, SVM classification algorithm was applied where training and testing datasets constructed by Mourad and Darwish [96] were used. The results showed that the highest performance on polar-vs-neutral classification was achieved by combining stem ngrams features with POS Tag features. The highest accuracy on 3-class classifications was 66.2%. The author has noticed that a rich set of morphological features is beneficial. On the other hand, she observed that the highest performance on positive-vs-negative classification was obtained by semantic features combined with stem ngrams features which has reached 82.70% accuracy.

In the research work of Al-Moslmi et al. [10], feature engineering was conducted, and the impact of the proposed set of features was examined. The researchers have extracted 26 features and grouped them into five groups. Six features were related to the frequency and the presence of the senti-words, seven features were related to the weight of senti-words, four features were related to POS tag, six features were related to sentence-level, with three other features. Most of the extracted features were semantic features as the developed lexicon was used to extract the semantic features. To examine the extracted features, five machine learning algorithms were applied to conduct 2-class classification. These algorithms were naive Bayes (NB), k-nearest neighbor (KNN), support vector machine

(SVM), logistic linear regression (LLR), and neural network (NEUNET). The results have showed that POS-based features and sentence-level features obtained high result compared to other type of features. POS-based features obtained the highest accuracy with 98% using the SVM classifier. On the other hand, LLR and NEUNET classifiers outperformed other classifiers with the use of all features with 97.8%, 97.6% accuracy.

Mourad and Darwish [96], as part of their research work, have extracted the following types of features: Stem-level features, Sentence-level features, Tweets-specific features and Language-independent features. Two lexicons were used to extract lexicon-related features. These lexicons were ArabSenti and a translated version of MPQA [47]. After designing and extracting the features, they have applied Naive Bayesian classifier. Two level-classification was conducted. They first conducted a subjectivity classification (polar-vs-neutral classification) followed by polarity classification (positive-vs-negative classification). First observation was that the stemming and POS tagging features have positive impact on the performance. Their results achieved 80.6% accuracy for polarity-classification and 77.2% accuracy for subjectivity-classification.

In a research related to sentiment analysis in multi-languages, Abbasi et al. [97] have evaluated syntactic and stylistic features for both Arabic and English contents. After extracting the features, four feature-selection techniques were examined: SVM weights (SVMW), Genetic Algorithm (GA), Information Gain (IG) and Entropy Weighted Genetic Algorithm (EWGA). EWGA was designed and developed to utilize both GA and IG. To evaluate the proposed selection techniques and extracted features, datasets of movie reviews and web forum posting were used. The SVM classifier was applied and the result

showed that EWGA outperformed other feature selection techniques. The authors have also reported that the stylistic features had a positive impact across all test datasets.

2.4 Classification

This section will discuss three levels of analysis and three approaches for classification. The levels of analysis are document level, sentence level and aspect level. In document-level classifications, the whole document is analyzed as a unit and is classified into its target polarity such as tweet and Facebook's comments. Sentence level-classifications deal with sentence as a unit for classification. The aspect-level classification needs more analysis to extract aspects. Aspect-level treats each attribute (better y, voice, etc.) of entity (iPhone) as a target for sentiment (good, bad, etc.). In the aspect-extraction, the attributes extracted from a review, then find the sentiment/opinion for each one. Each attribute called aspect and usually extracted with six objects: (Entity, attribute, opinion, time, opinion holders, orientations). For example, "*The iPhone's call quality is good, but its better y life is short*" assume that this sentence written by "Bob" at October 22, 2018 14:42:43. In this case, the example contains two aspects: one related to *iPhone's call* where the second related to the *better y life* which can be extracted like this: aspect1 ("iPhone's call", "good", "October 22, 2018 14:42:43", "Bob", positive) and aspect2 ("*better y life*", "*short*", "October 22, 2018 14:42:43", "Bob", negative) [1]. The aspect-level analysis performs more analysis to get more understanding about the orientation of each entity in the sentence. However, this required name entity recognition and aspect extraction.

There are three approaches to conduct classification: rule-based, learning-based and hybrid. In the rule-based, senti-lexicons were utilized along with some rules to classify the

given text into its orientation. For example, some researchers split the given text into terms, find them in the lexicon, score each term by lexicon score, apply negation rule to flip the orientation, and then accumulate the positive and the negative score to decide the target. For learning based, learning algorithms such as Naïve based are applied where the algorithms learn from historical data to construct classification models. Once the model parameters are determined and constructed, the model can be used to predict new data. The quality of the learning models depends on many factors such as the used data for learning, the used algorithm, features types, etc. The learning-based approach can be classified into two sub-approaches: shallow-learning and deep-learning [98]. NB, SVM, LR, K-NN, etc. are shallow-learning classifiers. Deep-learning classifiers can be Convolutional Neural Network (CNN), deep neural network (DNN), Long Short-Term Memory (LSTM), recurrent neural network (RNN), etc. [98]. The following Figure 2 illustrates the previous discussion.

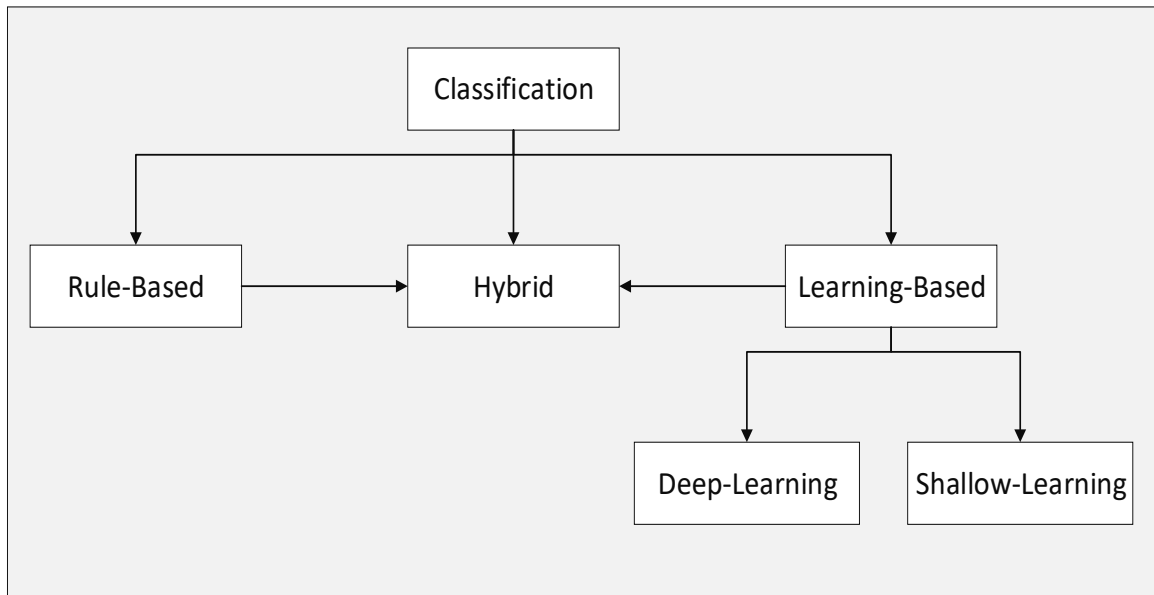


Figure 2. Classification Approaches

The following literature summary gives an overview of the impact of different classification analysis and algorithms.

In the research work of Abdulla et al. [81], two classification approaches were applied which were learning-based approach and rule-based approach. For learning-based method, SVM, NB, KNN and D-tree were examined. For the rule-based method, in-house tool was developed. The tool utilized the constructed lexicon to predict the sentiment on a given tweet. Their results have showed that the SVM outperformed others, and the learning-based approach performed better than the rule-based method.

In [74], Mountassir et al. have used SVM, k-NN and NB classification methods in document-level sentiment analysis. They have used three methods to address the problem of unbalanced datasets in sentiment classification and to examine the impact of imbalance datasets on classification. The results showed that 87.4% accuracy for SVM, 87.6% accuracy for NB and 76% accuracy for K-NN. The results also revealed that the SVM is highly sensitive to unbalance dataset, and the K-NN is less sensitive than SVM while NB was not sensitive.

In the investigation of Hammad and Al-awadi [99] of document-level sentiment analysis, four classification algorithms for supervised approach were examined. These algorithms were SVM, BPNN (Back-Propagation Neural Networks), NB, and Decision Tree. The dataset was collected on Jordanian hotels' customers' reviews from Twitter, Facebook and YouTube where 2000 informal Arabic reviews were collected. The results revealed 96.06% accuracy using SVM, 88.38% accuracy using NB, 69.77% accuracy using BPNN and 85.82% accuracy using decision tree.

Elouardighi et al [75], have applied SVM, random forests and decision trees for classification while investigating document-level sentiment analysis. The used dataset was collected from Facebook using Facebook Graph API on political comments where 10254 comments were collected. The proposed method focused on Modern Standard Arabic and Moroccan Dialectal Arabic. They have used different experiments with different features (unigrams, bigrams and trigrams) to decide which one was more appropriate. The results showed 81% accuracy for SVM, 77% accuracy for Random Forests and 75% accuracy for Decision Tree.

In the research work of Abuelenin et al. [100], document-level sentiment analysis was investigated. SVM and NB algorithms were used. The used dataset was collected from Twitter using Twitter's APIs on different topics in which 1560 annotated tweets was used for training, and a lexicon with 400 terms was used for annotation. The authors have reported 92.98% accuracy for SVM and 81.50% accuracy for NB.

Al-Moslmi et al. have conducted different comparative experiments to examine the quality and the usefulness of a developed Arabic senti-lexicon [10]. They have used document-level sentiment analysis and hybrid approach for classification. The researchers have used five supervised classification algorithms, which were Naïve Bayes (NB), K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Logistic Linear Regression (LLR), and Neural Network (NEUNET). The reported results showed 97.8% accuracy using Logistic Linear Regression (LLR) classifier.

Shoukry and Rafea in their research work on Sentence-level Arabic sentiment analysis [101], have used SVM and NB as classification algorithms. The used dataset was collected

from Twitter using Twitter APIs on different topics where 1000 tweets were annotated manually by three annotators. The results of sentiment analysis showed 73% accuracy using SVM and 65% accuracy using NB.

As an extension of the previous work, Shoukry has used a bigger dataset where 4600 tweets were annotated manually using three annotators [102]. SVM and NB classification algorithms were applied. The results showed that the increase of the corpus size lead to an increase in the accuracy. She has reached 77.7% accuracy for SVM, 74.6% accuracy for NB, 75.9% accuracy using lexicon and 80.9% accuracy for hybrid (SO+ML).

AlAsmar has used a hybrid approach to examine aspect-level sentiment analysis [103]. The lexicon and the ontology were used for unsupervised classifications. Decision Tree (DT), Naïve Bayes (NB) and K –Nearest Neighbor (K-NN) were used for supervised classifications. The used datasets were a public dataset about hotels, books, etc. The results showed that the book reviews had the highest accuracy, particularly for ontology with important features. The book reviews reached 72.75% accuracy using Decision Tree (DT), 70.35% accuracy using Naïve Bayes (NB), 76.15% accuracy using K –Nearest Neighbor (K-NN), 71.90% accuracy using Lexicon Baseline, 79.25% accuracy using ontology baseline and 80.15% accuracy using ontology with important features. The ontology was used to help aspects identifying and extracting.

Abd-Elhamid et al. have used lexicon-based classification while examining aspect-level sentiment analysis [104]. They have collected 200 reviews from various social media forms. Then, they have normalized and reorganized the rates associated with the reviews into one format (positive, negative, or neutral). They have focused on Modern Standard

Arabic. The main point is that they have proposed five rules to extract aspects and used the lexicon to evaluate the performance of each rule. The results have showed 92.15% accuracy.

ElSahar and El-Beltagy have used learning-based classification approach in their search work on document-level sentiment analysis [42]. Five shallow-learning algorithms were examined to validate the worth of their developed Arabic corpus and lexicons and to conduct hybrid classification approach. The algorithms were Bernoulli Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Linear Regression (LLR), and Stochastic Gradient Descent (SGD). They have explored different combination of features, classifiers and dataset setups, which lead to 3K experiments. The results have showed that the best classifier was SVM and the worst was KNN. They have pointed out that the document-length and the subjectivity richness have affected the sentiment classification.

In [105], Al-Azani and El-Alfy have used several learning-based supervised algorithms and techniques while conducting document-level sentiment analysis. They have examined the effect of words embedding word2vec feature on imbalance datasets. SMOTE (Synthetic Minority Over-sampling Technique) was applied to address the problem of imbalanced dataset in sentiment classification. The effect of unbalance vs balance dataset on sentiment classification was explored. The ensemble classifiers and single-based classifiers with different parameters were examined. For ensemble classifiers, voting, boosting, bagging, stacking and random forests were applied. For single classifiers, K-Neural Network (KNN), different kinds of Support Vector Machine (SVM), Logistic Regressions (LR), Stochastic Gradient Descent (SGD), Gaussian Naive Bayes (GNB) and

Decision Trees (DT) were examined. The results showed a good improvement when use SMOTE to balance the data. Furthermore, the average outcomes of ensemble classifiers performed better than single classifiers. However, the used dataset was limited to small size.

Al-Azani and El-Alfy evaluated several deep learning techniques on sentiment polarity classification through document-level sentiment analysis on Arabic microblogs [106]. Various forms of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) recurrent neural networks were applied. The extracted features were vectorized using word2vec neural language model. Two word2vec architectures were examined: continuous Bag-Of-Words (CBOW) and Skip-Gram (SG). The effect of the non-static models and the static models on several performance measures were evaluated. In addition, they have examined the influence of four optimizers on non-static CBOW models, which were Adagrad, Adam, Rmsprop and SGD. Their study showed that the LSTM performed better than CNNs models while the non-static models combined with LSTM performed better than static models. For the optimizers, their finding showed that the average performance of Rmsprop worked better.

In [107], Al-Azani and El-Alfy have applied document-level sentiment analysis with learning-based algorithms to evaluate the efficiency of using emojis features with other textual features on sentiment analysis. Five types of features were examined. These types were Term Frequency- Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA) with singular value decomposition (SVD), bag-of-emojis, word embedding using both Skip-Gram (SG) and Continuous Bag-Of-Words (CBOW). Additionally, the effects of Support Vector Machine (SVM) classifier using features selection techniques and

without using them were examined. For features selection techniques, they have used Correlation-based Feature Selection (CFS) with Best-First Search algorithm. Their findings showed that emojis features performed better than TF-IDF and LSA but not as CBOW and SG. Furthermore, the fusing textual features with emojis features has improved the results when the feature selection techniques were applied. However, over-dimensionality reduction on text sentiment analysis might affect the generalization of the model. For example, when they have applied feature selection on TF-IDF, they have reduced the dimensionality from 1762 to 50 features.

To conclude, most of the explored studies were applied on document-level classification specially Twitter documents (tweets). This might be because tweets have limited size and a tweet is close to a sentence-level classification. The length of the text plays vital roles in sentiment analysis where small size text has high probability to encapsulate one opinion about one entity. In the literature, most of the examined studies have reported that hybrid classifications can achieve high accuracy. The explored studies showed that deep-learning algorithms have performed better than shallow-learning algorithms. We observe that SVM and logistic regression achieved high accuracy in most of the study. Some results have reported that Naïve based classification was less sensitive to imbalanced dataset while SVM highly sensitive to imbalance datasets.

In this chapter, we reviewed the state-of-the-art about sports-fanaticism, building labeled senti-lexicons, building annotated corpora, the approaches of building classification model and features identification and extraction for sentiment analysis. In the next chapter, we will give an overview about our methodology.

CHAPTER 3

RESEARCH METHODOLOGY

Our research methodology has five phases. In the first phase, the concept of “*sport fanaticism in Text*” was formalized where indicators, criteria, and rules were defined and validated. In the next phase, we used this formalism as a foundation for building annotated corpora and senti-lexicons. In the third phase, several preprocessing tasks were applied. In the fourth phase, the process of identification and extraction features have been discussed. The fifth phase represents three classification approaches that have been applied in our work. Figure 3 illustrates the five phases of our research methodology, and the following description discuss each phase in details.

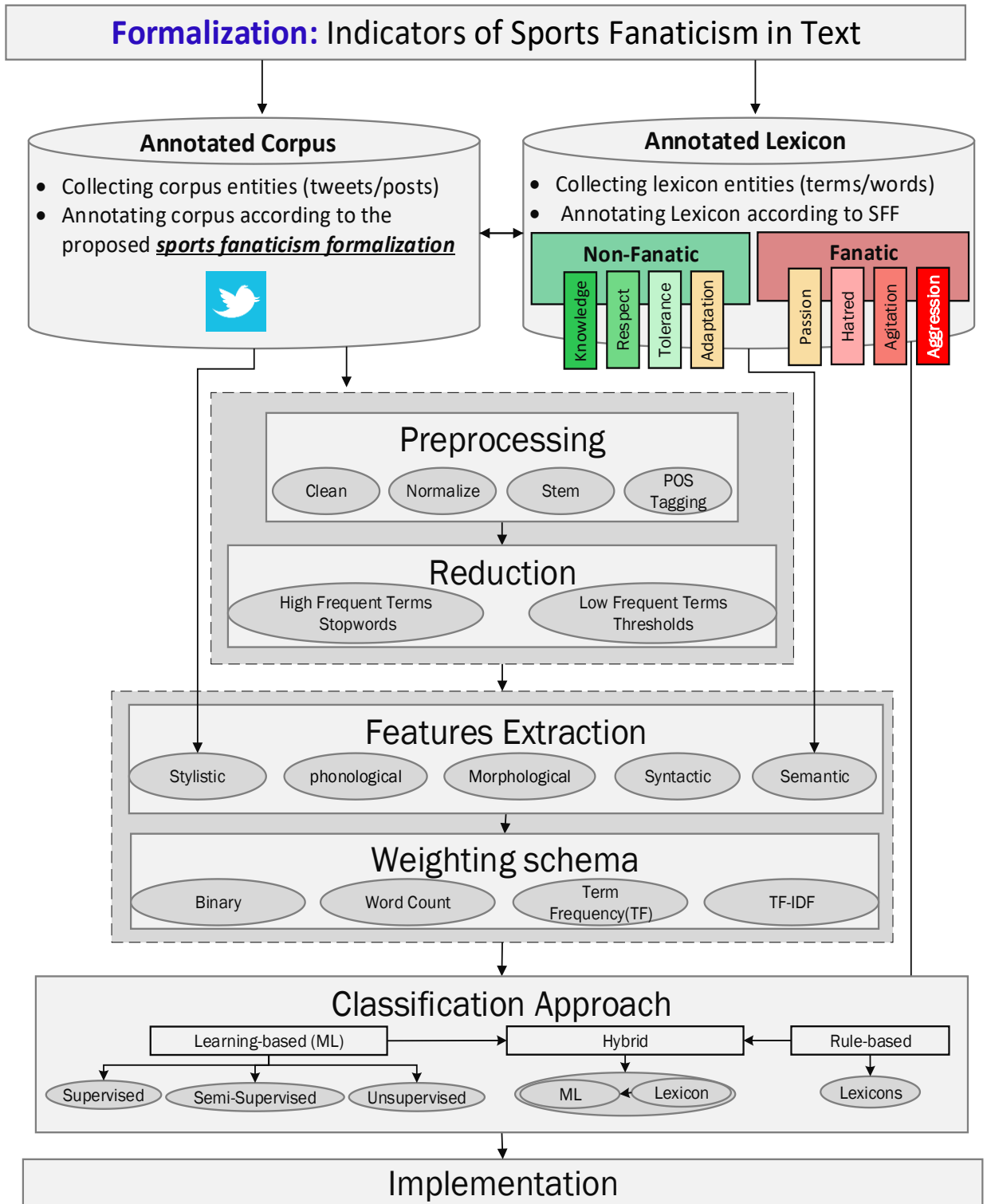


Figure 3. Research Methodology

In the first phase, we have attempted to answer the following two questions:

- What is the definition of “sport fanaticism in texts”?
- How to formalize the concept of "sport fanaticism in texts" into clear definition, criteria, rules and indicators for the purpose of building resources for sentiment analysis and classification models?

This phase started by understanding the phenomena of sport-fanaticism through reviewing state-of-art literature, analyzing many social media texts, and consulting a specialized expert. Then, brainstorming sessions were used to identify fanatic-related contexts. At this stage, we have identified 21 fanatic-related contexts. After that, the identified contexts were categorized into 8 categories (indicators/features). Then, these 8 categories were grouped into two classes: Non-fanatic (positive class) and Fanatic (negative class). After defining what each context is, the words and phrases of each context were collected. Therefore, a collection of words/phrases have been gathered. These words/phrases were used as prime-indicators to develop large-scale fanatic-related resources for Arabic sentiment analysis. We have called these words/phrases as *core fanatic-lexicon* that collected manually. The manual lexicon contains seeds of *senti-terms* for *sports-fanaticisms*. In addition, Chapter 44 provides an extensive description of *fanatic-formalism*. We have called the output artifact of this phase as *Sport fanaticism formalization* where the concept of *sports-fanaticism* is formalized into contexts, categories and indicators. We have spent enough time to develop and validate the formalism as it is the vital foundations of other stages of this research work. An expert specialist in social psychology of sports has helped in validating our proposed formalism. As far as we know, this is the first work to formalize

the concept of *sport-fanaticism in text* with purpose of automating the process of detecting and classifying social text into fanatic and non-fanatic. We believe that this work will be a valuable product for society, especially in Saudi Arabia. It will help as a guide to build applications that work as anti-Fanaticism. It will also help as a guide to measure the degree of fanaticism in the social media. This will help the governments to measure the impact of their efforts to reduce this phenomenon.

In the second phase, the required resources to design and develop a classification-model were built. The formalism developed in the first phase was used as guide to build the needed resources. The built resources have included: annotated corpora, senti-lexicons, language models, and dictionaries. First, 3.5M sport-related tweets were collected using Twitter APIs [108]. Then, the collected tweets were cleaned and preprocessed. The cleaned dataset has been annotated using the indicators and fanatic-terms developed in first phase. In other words, the developed indicators were used as seed words to annotate the collected tweets as *fanatic* or *non-fanatic*. The annotation has been done automatically. Hence, the annotation process has resulted in a large-scale annotated corpus with around 300K annotated tweets. This corpus was used to extract several domain-related lexicons using statistical and association measures. A generic sport-related language-model was also built and used as a utility to build related lexicons. Many other dictionaries were also collected. Some of these dictionaries were negation terms and patterns, modal words, intensifier words and other sentiment analysis dictionaries. To build senti-lexicons, we have utilized two different approaches: PMI and adjusted-TFIDF. We proposed and developed an algorithm to detect important fanatic-words and automatically score them. In this phase, we have also constructed new annotated corpus with around 10K tweets. This corpus was

annotated semi-automatically where we developed web-app to simplify the process of annotation. Enough efforts were put to prepare and validate this corpus. We have used this corpus as a testing dataset. As a result, we have a training dataset of 300K tweets and a testing dataset of 10K tweets. Building resources for sentiment analysis requires developing several programs and tools to collect data, to clean and preprocess them, and for data annotation. As far as we know, there are no sport-fanatic resources available for sentiment analysis. We believe that the developed resources will be useful resources for the research community working in Arabic sentiment analysis. We provide extensive details on the constructed resources in Chapter 7.

In the third phase, preprocessing tasks were applied. Preprocessing tools for Arabic text play a vital role in sentiment analysis where they form the data to be suitable for classification. In the third phase, we have utilized some of available tools, adjusted some other tools, and developed our own tools for preprocessing tasks. To clean the data, a cleaner program was developed to handle the following tasks: removing non-Arabic characters, removing URL, removing digits, removing punctuations, removing duplicate spaces, removing lengthening characters, etc. A normalizer and a tokenizer were also developed. For stemming, segmentation and lemmatization we have used Farasa stemmer [109]. More details about preprocessing tasks are covered in Chapter 6.

In the fourth phase, feature engineering tasks were conducted to convert the raw data into suitable input data for classification model. The fed input for classification model consists of features, attributes, or variables. The tasks of feature engineering are feature-identification, feature-construction, and feature-reduction. In feature-identification, understanding the domain-problem and analyzing the data were mandatory to identify

useful features. As a result, five feature-sets were identified. These features are: stylistic features, phonological features, Morphological features, syntactic features, and semantic features. The identified feature-sets revealed both container-related features (stylistic feature-set) and text-related features (remaining four feature-set). In addition, they exposed features from low-level (phonological feature-set) to high-level (semantic feature-set). The proposed features can play vital roles for Arabic sentiment analysis. We have examined the use of these features and provided details in Chapter 8. After identifying features, Java programs were developed to extract features. Then, we have pursued feature-selections through Weka [110] using chi-square technique (called in Weka: ChiSquaredAttributeEval).

In phase five, we have used sentiment analysis and machine learning algorithms to design and develop the needed classification models. The proposed classification-models were designed to classify the text into fanatic or non-fanatic sentiment/opinion. There are three classification approaches: learning-based approach, rule-based approach, and hybrid approach. We have applied all three approaches. For rule-based approach, the developed lexicons for sport-fanatic were used to classify a given tweet by calculating the score of each word and subtracting the non-fanatic-related weight from the fanatic-related weight. The developed lexicons show high accuracy. For learning-based approach, all feature-sets that were extracted in feature-engineering phase were examined. We have applied SVM, LR and NB classification algorithms for each feature-set. Then, we have examined the combinations of different feature-sets. The experiments represent the impact of the identified feature-sets on classification. For hybrid approach, both learning-based and rule-based approaches were combined. We have applied three scenarios. In first scenario, we

have selected one lexicon and conduct lexicon-based classification. The lexicon-based classification has resulted into eight weights, one for each category (indicator/features) in our formalism discussed in Chapter 4. Then, we have used the output of the lexicon-based as an input to the learning-based algorithms such as SVM, LR and NB. Therefore, the input to learning-based algorithms contained eight features. The hybrid approach showed promising results.

After applying these five phases of research methodology, we have developed and implemented a web service to discover the sport fanaticism in text automatically.

In this chapter, an overview of the research methodology was given where five phases were described. These phases are concept formalization, building resources, preprocessing, feature engineering and classification. In the next chapter, the concept of sport fanaticism in text will be presented. Concept will be formalized into indicators that help to build fanatic resources.

CHAPTER 4

SPORT FANATICISM FORMALIZATION

This chapter concentrates on formulating the concept of Sports-Fanaticism in Text (SFT) for Arabic Sentiment Analysis (ASA). Sports-fanaticism is one of the social problems that have fanatic impacts on the social fabric. The sports-fanaticism is a psychological emotion with hate against the opposing team and love for own team. This emotion is not controlled by mind but by sentiment. The sport fanaticism is a fanatic trait, especially if it becomes a social phenomenon. Social network sites have become a mean for communication between sport fans. Therefore, we need a tool that helps in automatically detecting and measuring this phenomenon in social network sites such as Twitter and Facebook.

The aim of this part of the research work is to formulate the concept of *sports-fanaticism in text* into criteria, rules and indicators/features. The proposed formulation can be used in Arabic Sentiment Analysis to automatically detect Sports-fanaticism in social text.

Nowadays, social media sites become common means for sport fans to communicate [7]. Therefore, understanding how social text contributes to increase or decrease sports-fanaticism is essential. The formalization of sport fanaticism will help to automate anti-fanatic and build measurement-tools to assess text quality from the perspective of sports-fanaticism.

In the following sections, we proposed indicators/features that help classifying social text into fanatic and non-fanatic. The non-fanatic text helps to decrease sports-fanaticism where

the fanatic text increases sports-fanaticism. This formalization is a foundation for feature engineering stage of this research work. It provides a description of domain-knowledge that is needed to understand the domain-problem. Understanding the domain-problem helps to convert the given text into suitable attributes for machine learning models.

4.1 Formalizing Sport Fanaticism into Indicators/Features

In this section, we attempt to handle the following questions: What is the definition of “*sport fanaticism in texts*”? How to formalize the concept of “*sport fanaticism in texts*” into clear definition, criteria, rules and indicators for the purpose of building sentiment analysis resources and classification model?

Fanaticism appears in different domains such as religions and politics. In this research, we will focus on fanaticism in sports domain. We will call our domain “sports-fanaticism”. Alshehri et al [13] translated the Allawi’s [111] definition about sports-fanaticisms as "blind hatred of the members of competing sports teams combined with a blindness toward the faults of the individual’s favored team. The individual’s emotional identification with their favored team causes them to overlook obvious facts that their team may be less than perfect or that a competing team may have certain virtues”. Hence, we can restate the sports-fanaticism as a psychological emotion with extreme hatred against the competing team and excessive love for her/his team.

For “*sports-fanaticism in texts*”, we propose the following definitions for both fanatic-text and non-fanatic-text. See Figure 4 and Figure 5.

Definition 1

Fanatic-text is a text that explicitly or implicitly contains aggression, agitation, hatred or/and passion.

Figure 4. Definition of Fanatic-Text

Definition 2

Non-fanatic-text is a text that explicitly or implicitly contains adaptation, affection, respect or/and knowledge.

Figure 5. Definition of Non-Fanatic-Text

Accordingly, we will focus on eight concept-indicators (features), four of which increase sports-fanaticism which are: aggression, agitation, hatred or passion, and four indicators that reduce sports-fanaticism which are: adaptation, affection, respect and knowledge. In the next sections, we split each indicator into more precise meaning that represents a specific context. Then, we link each context with possible words and phrases that are appropriate for it. Consequently, the concept of sports-fanaticism in text is formalized into indicators, contexts and phrases. This will give a clear conception of the concept of *sports-fanaticism in texts*. In an innovative manner, the concept can make sense when it is formalized to be suitable for sentiment analysis. The concept is sequenced from the definition into concrete details, including words and phrases. In the following subsections, we will present the concept as a hierarchy (concept-indicators -> contexts -> words and phrases). Figure 6 illustrates the proposed formalism.

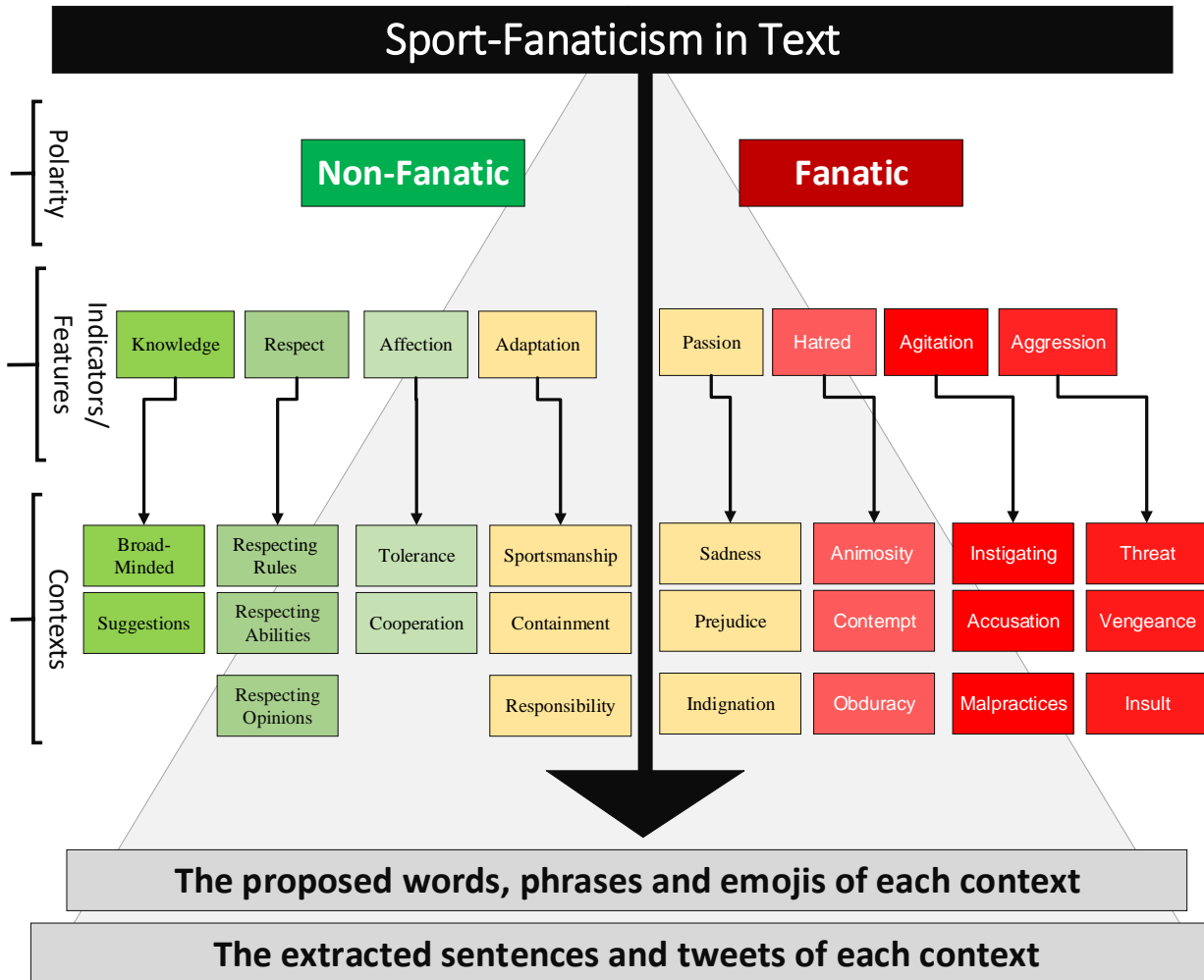


Figure 6. Overview for Sport Fanaticism Formalization (SFF)

4.1.1 Fanatic Indicators

Our proposed definition encompasses four fanatic indicators: aggression, agitation, hatred and passion. Each indicator involves several contexts where each context can be detected using numerous words and phrases. The following subsections describe in detail these four fanatic indicators, indicators' contexts, and the proposed Arabic words and phrases belong to each context.

4.1.1.1 Aggression

Aggression could be considered as an extreme-fanatic indicator where the fanatic sentiment is transformed to an action that hurts others. This action can be expressed in the following contexts: threat, intimidation, insulting, and revenge. These contexts and their representative words and phrases are described as follow:

Threat and intimidation: This is a context that implicitly or explicitly contains threatening (menacing) expressions, such as threatening to attack, threatening to violate (disrupt, infringe), threatening teams, fans, players, referees, linesmen or club management, riot encouragement, attacking encouragement, quarrel and assault, showing a willing to attack others, or embodiment of hostility. Table 3 contains some Arabic words and phrases that are often appear in this context. Figure 7 presents an example statement extracted from Twitter in this context.

Table 3. Some words and phrases for the context of threat

يجلد	يرجم	يذبح	يدعس	يحرق
يقتل	يضرب	يصلب	يشنق	يسجن

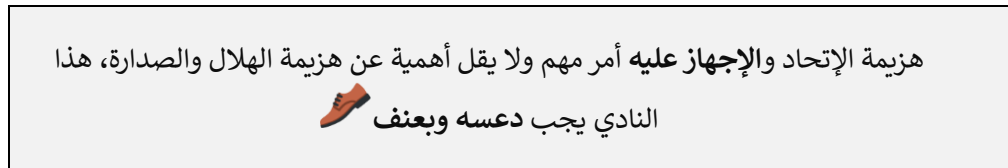


Figure 7. An example of threat context

Insult: It is a context that implicitly or explicitly contains insulting expressions such as insulting using father's reference (e.g. يل*ن ابوكم) or any other family's reference. It might contain insulting using animal analogy (e.g. حم*ر donkey), insulting using name-calling (e.g. الفقراوية), insulting using injurious or humiliating remarks, insulting using abusive phrases, insulting using libel, slander and denigration (e.g. عبد وضيع rascal). It might also contain insulting phrases to competing/opposing teams, fans, players, referees, linesmen or club management. Table 4 contains some Arabic words and phrases that are often appear in this context. Figure 8 presents an example statement for this context extracted from Twitter.

Table 4. Some words and phrases for the context of insult

أهبل	حيوان	بن كلب	الانجاس	يا بن الجزمة
بن الشر_KWS_ة	وسخ	مختل عقلياً	صعلوك	سافل

ي مووووووت قهر، نادي معفن وجمهور بن كلب

Figure 8. Example of insult context

Revenge: This is a context that implicitly or explicitly contains expressions of revenge, such as promising to vengeance, requests for retaliation, nasty warning, commination, jeremiad, menace or phrases that imply fanatic payback in future towards teams, fans, players, referees, linesmen or club management. Table 5 contains some Arabic words and

phrases that are often appear in this context. Figure 9 presents an example of such context extracted from Twitter.

Table 5. Some words and phrases for the context of vengeance

الدم يغسل بالدم	الثأر ولا العار	الثأر الثأر	البادئ أظلم	الانتقام
تحملوا ألى سوف يحدث	لن ننسى	سنثأر بقسوة	يجب ان نثأر	السن بالسن

يوم الثأر والانتقام الرهيب ل هذا النادي الرديء، ان شاء الله ب التأكيد سوف نرد الصاع صاعين

Figure 9. Example of vengeance context

4.1.1.2 Agitation

Agitation is a synonym of incitement, instigation, agitation, sedition, incitation and incendiary. It is a fanatic indicator that implicitly or explicitly contains contexts (phrases or symbols) that lead to sedition. It involves the following contexts: instigating fans, accusation and promoting malpractices. The next subsections present more details about these contexts.

Instigating fans: It is a fanatic context that implicitly or explicitly contains expressions or symbols of overcharging and jeering fans such as rude remarks, mocking remarks, hostile rhetoric, overcharging against the competing/opposing team, remarks that make the loss unacceptable, inciting against the competing/opposing fans and inciting of coarseness action against competing/opposing teams, fans, referees, players, linesmen and club

management. Table 6 contains some Arabic words and phrases that are often appear in this context. Figure 10 presents an example of this context extracted from Twitter.

Table 6. Some words and phrases for the context “instigating fans”

لن نقبل الهزيمة	اقهروهم	قاطعوا	المقاطعة	لن نقبل الهزيمة
الحكم معاهم	الاتحاد ضد النادي	الحكم ضدنا	يسقط	الحكم معاهم

هذا حساب الصحيفة _____ وهذا مروجها _____ ارجموه بالبلوك، ليعلم قيمة
الملكي وجماهيره
#...! صحيفة_النادي_تسيء_لجمهور_الملكي

Figure 10. Example of the context “instigating fans”

Accusation: Accusation is any fanatic context that implicitly or explicitly contains expressions or symbols of an action of damaging the good reputation of competing/opposing teams, fans, referees, players, linesmen and club management. For example, charging club president with theft, charging a referee with treason or other types of defamations and slanders without any evidence. Table 7 contains some Arabic words and phrases that are often appear in this context. Figure 11 shows an example of this context extracted from Twitter.

Table 7. Some words and phrases for accusation context

إرهابي	المرتزقة	العملاء	الصهاينة	الخونة
نصاب	منافق	متآمر	ماسوني	لص



Figure 11. An example of accusation context

Promoting malpractices: it is a fanatic context that implicitly or explicitly contains expressions or symbols that encourage fanatic habits. These are contrary to sports ethics, values and norms, or promoting activities. It also contains terms incompatible with national culture. For example, encouraging lies, encouraging fraud, encouraging fallacies, encouraging unlawful behavior, encouraging to throw fans of the competing/opposing team by objects, encouraging willful misconduct, advising to do bad practice or discouraging good practice. Table 8 contains some Arabic words and phrases that are often appear in this context. Figure 12 presents an example of this context extracted from Twitter.

Table 8. Some words and phrases of promoting malpractices context

سيبك من اللوائح والقوانين	حاول تفشله	لا تحترموا	أتحرش بالمشجعين	خالف تُعرف
أتركك منهم	اكسره	ابسق على وجهه	اكذب عليه	خر الوقت

اللاعب اللي ما توقفه اكسره 🖐️ - معترض _____ #السعوديه_بلجيا

Figure 12. An example of the promoting malpractices context

4.1.1.3 Hatred

Hatred is any fanatic-indicator that implicitly or explicitly contains contexts of animosity, racism, schadenfreude, mockery, and inflexible. The following subsections present more details about these contexts.

Animosity or racism: it is a fanatic context that implicitly or explicitly contains expressions or symbols of hating other teams, fans, referees, players, linesmen and club management. It contains phrases that imply ethnic, nationalism and racial discrimination. It includes phrases that indicate malice, jealousy, envy and segregation such as a willing to cause harm to others. Table 9 lists some Arabic words and phrases that are often appear in this context. Figure 13 presents an example of this context extracted from Twitter.

Table 9. Some words and phrases of animosity and racism context

حسود	اغث نادي	عبد	بغيض	حقير
مكروه	دخيل	أجنبي	قبيح	قدر

ما كرهت للاعب زي غالب كرهه الله يأخذه 🤔

Figure 13. An example of animosity and racism

Contempt (Schadenfreude and mockery): it is any fanatic context that implicitly or explicitly contains expressions or symbols of blasphemy, contempt and ridicule. Such context has provocative words, expressions of arrogance or boasting. Some examples are: expressions of joy when the competing/opposing team is given a warning or red-card to one of its players, the irony of the decisions of the referee, expressions of joy and healing when a member of the competing/opposing team is injured and moves out of the field, words derogating from other clubs, contempt or derogation of fans of the competing/opposing team, or words that provoke the competing/opposing team. Table 10 contains some Arabic words and phrases that are often appear in this context. Figure 14 presents an example of this context extracted from Twitter.

Table 10. Some words and phrases of contempt context

غبي	أحمق	ضفادع	طحالب	الطواقي
دنيء	بطيخ	نادي الشحاة	مقرف	تافه

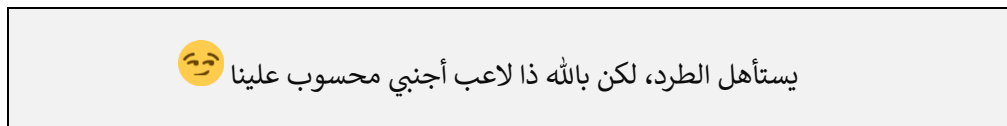


Figure 14. An example of contempt

Inflexibility or obduracy: A fanatic context that implicitly or explicitly contains expressions or symbols of non-acceptance of the views of others, absence of apologies, lack of containment statements after criticism, expressions of disrespect for others, expressions of disrespect for others or disrespect other points of view. Table 11 contains

some Arabic words and phrases that are often appear in this context. Figure 15 presents an example of this context extracted from Twitter.

Table 11. Some words and phrases of obduracy

لا نقبل بغير	خط أحمر	انتهى النقاش	الأعظم	لن نقبل اعتذارهم
لا تصالح	لا تعون	لا تفاوض	لا سلام	لا شرکه

الاهلي كبير جده رغم انفك، الكبير كبير والصغير صغير، والعين ما تغلا ع الحاجب، ف
الخدم يضلون خدم لا يمكن رفعهم بمرتبة سيد حتى لو عنده 2 او 3 آسيا..

Figure 15. An example of obduracy

4.1.1.4 Passion

A fanatic-indicator that implicitly or explicitly contains contexts of extreme sadness, extreme happiness, extreme prejudice and indignation. The following subsections present more details about these contexts.

Extreme sadness: A fanatic context that implicitly or explicitly contains expressions or symbols of discontent, objection to the outcome of the match, crying, extreme grief, and fury if any player is suspension or expulsion. Extreme-sadness might also be sensed by the availability of nervousness if any player substitutes by other, extreme anger and extreme indignation about injury to a team star or team loss, objection or stern protest, excessive blaming, excessive resentment, excessive discontent, prayers against others, swear or the

fear of defeat. Table 12 shows some Arabic words and phrases that are often appear in this context. Figure 16 presents an example of this context extracted from Twitter.

Table 12. Some words and phrases of extreme-sadness

لا يطاق	فاجعة	أف	اكتئاب	اضجر
يا حسرتاه	منحوس	مؤلم	قهر	مأساة

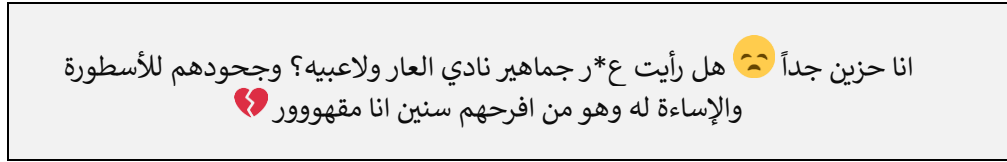


Figure 16. An example of extreme-sadness context

Happiness with prejudice: A fanatic-indicator that implicitly or explicitly contains expressions or symbols of exaggerated blind love, pretense of perfection, personality, exclusivity, self-love, selfishness, show off, boasting, overstatement, over-praise, differentiation, defending about mistakes, excess of courtesy or favoritism, bullying, pray and swear. It contains expressions or symbols of excessive joy that provokes a grudge in the hearts of the fans or showing excessive happiness because the falling of the opponent. Table 13 contains some Arabic words and phrases that are often appear in this context. Figure 17 presents an example of this context extracted from Twitter.

Table 13. Some words and phrases of extreme happiness with prejudice context

العالمي	الأسطورة	الأفضل فريق	الأعظم فريق	قسم بالله
ما أعظم هذا الكيان	الأفضل في الكون	النادي التاريخي	نادي الشعب	نادي الوطن

#الاهلى_فوق_الجميع: نحن جمهور الكيان، نحن جمهور الأعظم نادي في أفريقيا، من انت؟! #ادعم_الاهلي

Figure 17. An example of extreme happiness with prejudice

Indignation: Any fanatic-indicator that implicitly or explicitly contains phrases or symbols of objection to the referees, objection to the decisions of the referee, excessive criticism against referee/coach, pray, or swear against the competing/opposing teams.

Table 14 contains some Arabic words and phrases that are often appear in this context.

Figure 18 presents an example of this context extracted from Twitter.

Table 14. Some words and phrases of indignation

يارب يفشل	ورب الكعبة	والله اكتتاب	بالله فكونا	الله لا يوفق
جعله يهبط	أحرق قلوبنا	الله يحرقكم	يارب يهبطون	الله يأخذهم

بالله هذا حكم؟! خرب المباراة، الله يأخذه

Figure 18. An example of indignation

4.1.2 Non-fanatic Indicators

Our proposed definition encompasses four non-fanatic indicators which are: adaptation, affection, respect and knowledge. Each indicator involves several contexts where each context could be detected using numerous words and phrases. The following subsections describe the four non-fanatic indicators, indicators' contexts, and the proposed Arabic words and phrases belong to each context.

4.1.2.1 Adaptation

Adaption is a non-fanatic-indicator that implicitly or explicitly contains context of sportsmanship, containment or responsibility. The following subsections present more details about these contexts.

Sportsmanship: A non-fanatic context that implicitly or explicitly contains expressions or symbols of fair play, expressions that indicate moral and civilized cheerleading, expressing the happiness in words that do not provoke grudges, or expressions of the noble winner's ethics. Table 15 contains some Arabic words and phrases that are often appear in this context. Figure 19 presents an example of this context extracted from Twitter.

Table 15. Some words and phrases of sportsmanship

ألف مبروك للفائز	شجعنا لاعبيننا	روحنا رياضية	رائعة	شكراً للفريقين
حض او فر للخاسر	أدى جهده	يستأهل الفوز	هرد لكم	مباراه رائعة

ألف مبروك لكل محبين النادي الأهلي، وهاردلك لكل محبين نادي الهلال، اهم شى تكون
روحنا رياضية #الهلال_الأهلي_نهائي_ولي_العهد #الاهلي_الهلال

Figure 19. An example of sportsmanship

Containment: Containment is a non-fanatic context that implicitly or explicitly contains expressions or symbols of helping making things calm down, words to ease tensions of debate, phrases that reduce the emotional stress of the losing team, or containment statements after criticism. Table 16 contains some Arabic words and phrases that are often appear in this context. Figure 20 presents an example of this context extracted from Twitter.

Table 16. Some words and phrases of containment context

اداء رجولي	القادم أفضل	لم يحالفه الحظ	لأسباب نجهلها	أدى جهده
كلنا شعب واحد	كنتوا الاجدر	يعطيكم العافية	بالتوفيق بالقادات	أهم شي المتعة

كل الشكر للإخواني اللاعبين على هذا المستوى الرائع الذي قدموه في المباراة، روح عالية
لأخر دقيقة، ولكن الركلات الترجيحية لم تبتسم لنا للأسف، وقدر الله وما شاء فعل، كل
الشكر للجماهير الحاضرة فعلاً كنتوا الرقم 1 في المباراة وبإذن الله القادم أجمل ❤️

Figure 20. An example of containment

Responsibility: A non-fanatic context that implicitly or explicitly contains expressions or symbols that indicate constructive dialogue, indicate self-critical, or indicate self-restraint. It contains expressions of taking responsibility or apologizing for mistakes, statements of accepting right things, or phrases of accepting excuse. Table 17 contains some Arabic

words and phrases that are often appear in this context. Figure 21 presents an example of this context extracted from Twitter.

Table 17. Some words and phrases of responsibility

نثق فيكم	نبدل	تركيز أكثر	المحاولة	الأخطاء واردة
نحو الهدف	نحثهم	لا يتخاذلوا	لا يأس	هناك مشكلة

نادي الاتفاق منظم من جميع النواحي، والكل مندمج في العمل وحريص على الأفضل، وأنا سأعمل على تحقيق طموح جماهيره والوصول لنتائج إيجابية

Figure 21. An example of responsibility context

4.1.2.2 Affections and Intimacy

This is an non-fanatic-indicator that implicitly or explicitly contains contexts of tolerance, cooperation or flexibility. The following subsections provide more details on these contexts.

Tolerance: Tolerance is a non-fanatic context that implicitly or explicitly contains expressions or symbols of friendliness, pleasantness, friendship and tenderness. For example, using expressions of greetings, tributes, salutations, compassion, love, brotherhood, peaceful coexistence, apology, and forgiveness. Table 18 contains some Arabic words and phrases that are often appear in this context. Figure 22 presents an example of this context extracted from Twitter.

Table 18. Some words and phrases of tolerance

ألف سلامة	ألف مبروك	الله يحييك	الله يخليك	أخوي
يا غالي	تشكرون على	العزیز	الله يديم الأفراح	الحمد لله

شكراً اخي نادر، نادي مرباط راح يكون رقم صعب في الدوري والكأس بإذن الله

Figure 22. An example of tolerance

Cooperation: Cooperation is a non-fanatic context that implicitly or explicitly contains expressions or symbols of helping each other, understanding each other, and/or indicating teamwork. It contains expressions that show good communication with each other or willingness to assist. Table 19 contains some Arabic words and phrases that are often appear in this context. Figure 23 presents an example of this context extracted from Twitter.

Table 19. Some words and phrases of cooperation

التلاحم	التكامل	التكافؤ	الانسجام	الاندماج
نعمل سوياً	لجميع	لبعضنا البعض	التكاتف سوياً	التعاقد


رئيس نادي الاهلي: إذا أردنا بطولة الدوري علينا التكاتف سوياً لتحقيقه قضايا الاهلي الدولية والداخلية بسيطة وقليلة 

Figure 23. An example of cooperation

4.1.2.3 Respect

It is a non-fanatic-indicator that implicitly or explicitly contains respect related contexts. This includes contexts respecting rules and regulations, respecting others' abilities, respecting others' beliefs and opinions, respecting others' identities and affiliations. The following subsections provide more details on these contexts.

Respecting rules and regulations: This context is a non-fanatic context that implicitly or explicitly contains expressions or symbols of referee decisions respect, linesmen decisions respect and FIFA respect, respect expression for game's rules, or contrary's culture and laws. It might also contain respect expressions of the decisions made by club management. Table 20 provides some Arabic words and phrases that are often appear in this context. Figure 24 presents an example of this context extracted from Twitter.

Table 20. Some words and phrases of respecting rules

احترام القرارات	احترام القوانين	الالتزام بالقوانين	التحقيق	الإجراءات
القانون	اللائحة	القوانين	الإجراءات القانونية	حسب قرارات

أُتقبل القرار الصادر بكل احترام، وسأواصل خدمة الرياضة السعودية من خلال نادي الهلال مع احترامي وتقديري للجميع

Figure 24. An example of respecting rules

Respecting Others' abilities: This context is a non-fanatic context that implicitly or explicitly contains expressions or symbols that reflect respecting the opposing/competing

teams and their abilities. Table 21 lists some Arabic words and phrases that are often appear in this context. Figure 25 presents an example of this context extracted from Twitter.

Table 21. Some words and phrases of respecting others' abilities

لا نستنقص من أحد	المنافس	عدم الاستهانة	احترامنا للجميع	اتمنى ان تحترم
لا نسيء إلى	حاول الاجتهاد	قدم كل ما يملك	قدم كل ما يستطيع	قدرات مميزة

مشكلة نادي الشباب هي الاستهانة بالمنافس.. احترام قدرات الخصم والتركيز الكامل بالمباراة
ممکن يعطي فارق كبير.. هارد لك لكل الشبابيين

Figure 25. An example of respecting others' abilities

Respecting others' beliefs and opinions: A non-fanatic context that implicitly or explicitly contains expressions or symbols of respecting others' points of views, respecting others' feelings or respecting others' ideas. Table 22 contains some Arabic words and phrases that are often appear in this context. Figure 26 presents an example of this context extracted from Twitter.

Table 22. Some words and phrases of respecting others' beliefs and opinions

لكل شخص وجهة نظر	اتمنى ان تحترم	نحترم رأيك	رأي شخصي	من وجهة نظري
احترام قرارات الحكم	لك التقدير أخوي	يستحق الاحترام	وجهة نظر	هذا رأي شخصي

استاذ _____ نحترم رايبك، يبقى الاحتجاج القانوني حق مشروع لكل نادي، وإلا اصبح القانون واللوائح لا قيمة لها ..

Figure 26. An example of respecting others' beliefs and opinions

4.1.2.4 Knowledge

Knowledge is a non-fanatic-indicator that implicitly or explicitly contains contexts of ideas, solutions, suggestions, intellectual thought, visions or other useful information. The next two subsections provide some details on these contexts.

Broad-minded: This context is a non-fanatic context that implicitly or explicitly contains expressions or symbols of the knowledge of regulations and laws. It contains expressions of the true meaning of sport, expressions of awareness of the noble objectives of sports, or expressions that increase the awareness of the benefits of athletic activities. Table 23 contains some Arabic words and phrases that are often appear in this context. Figure 27 presents an example of this context extracted from Twitter.

Table 23. Some words and phrases of broad-minded context

من اللائحة	في الحالات التالية	في الفقرة	في الفقرة أعلاه	عندما تكون الكرة
في المادة من	في المباريات التي	في المنطقة التقنية	سيطرة على الكرة	في النظام الاساسي

مجلس الأهلي يقرّ تعديلات بعض بنود مشروع لائحة النظام الأساسي مجلس إدارة النادي الأهلي، برئاسة الكابتن محمود الخطيب، قرر في اجتماعه الذي جرى ليلة أمس، الاثنين، مناقشة العديد من المقترحات التي تقدم بها أعضاء الجمعية..

Figure 27. An example of broad-minded context

Suggestions: A non-fanatic context that implicitly or explicitly contains expressions or symbols of providing ideas, creating visions, providing useful advising, or helping in finding solutions. Table 24 contains some Arabic words and phrases that are often appear in this context. Figure 28 presents an example of this context extracted from Twitter.

Table 24. Some words and phrases of suggestion context

من التحديات	من البدائل	من الأهداف	من الأفكار	من الإيجابيات
من العوامل	من الصعوبات	من السلبيات	من الرؤى	من الحلول

من الحلول وضع مكافأة ماليه مجزيه للأندية عن كل فوز يحققه اي نادي بالدوري

Figure 28. An example of suggestion context

CHAPTER 5

DATA COLLECTION

In this chapter, we describe the data collection phase. We have used Twitter [112] as the source of our dataset. Twitter is a social media site where people can share their sentiments, opinion and ideas with others. We have selected twitter site to be our data source due to several reasons. First, it is a reach of our target sentiments information where Saudi sport fans use it frequently to deliver their sentiments. As reported by SemioCast [113] in 2012, the Riyadh had reached the 10th active city on twitter and Arabic usage is continuously growing. Secondly, Twitter provides streaming features to collect the whole tweets. Many Twitter APIs in different programming languages are available and free to use. Thirdly, Twitter is appropriate for sentiment analysis because the size of each tweet is limited to 400 characters where sentiment and target labels can be detected easily. Twitter provides and supports three levels of data access through three packages [114] that allow researchers or users to collect historical data. These packages are standard, premium and Enterprise packages. It also provides steaming features to collect real-time tweets. The Standard package is free and allows the user to search for tweets in last 7-days. premium package can be used through a formal contract and approval where Twitter allows the user to search for tweets in last 30-days or full-archive since 2004. For streaming APIs, Twitter API platform offers streaming real-time tweets. It allows a number of filters where you can use up to 400 keywords for string search. It also provides a search using user ID. It allows to search using 5,000 user_IDs. You can search also using 25 location boxes. There are many

libraries in different programming languages that can be used to connect to Twitter services and conduct streaming. We have used Hosebird Client (HBC) [115]. HBC is a Java HTTP client for consuming Twitter's standard Streaming API.

To store the collected tweets, we have used MongoDB [116] database. It is a NoSQL database. NoSQL databases are fast and appropriate for many situations. NoSQL refers to non-relational database. Therefore, the overhead of relational database is avoided. This provides a fast and a fixable data storage. NoSQL databases have a positive impact on big-data and real-time applications. MongoDB stores data in key-value data-store type. It stores the data in JSON format. The table in MongoDB database is called a collection while the row data is called a document. We have found that MongoDB is suitable as a datastore for the collected tweets due to several reasons. The received tweets from Twitter streaming services are JSON format objects. Therefore, they can be inserted directly to the MongoDB. MongoDB provides an engine to easily query the collected tweets using different criteria. It provides Java and Python APIs to access the collected data where we can easily use them when we need to retrieve the data for preprocessing and machine learning. We highly recommend it to who likes to work in Arabic processing and machine learning. MongoDB helps to get the advantages of centralizing the data and access them from different programming languages easily using APIs. For MongoDB GUI, we have found that the best tool is Studio 3T [117].

The received object from Twitter streaming APIs is a JSON object. It has a key-value structure where the key represents the name of the field, and the value represents the data of the field. Figure 29 shows an example of one of the received tweets using Tweets API. It contains several attributes such as the text of the tweet, creation time, user information,

language of the tweet, replay to whom, truncate or not, retweet or not, and several other metrics. Truncated field/attribute in the collected tweet is important field to us. It helps when extracting the text of the tweets. If the truncated field is false it means that *field text* contains the whole tweet's text, otherwise, the *field text* does not have the whole text and it truncated. Therefore, the whole text can be found in another field called *extended tweet*. This happens because the *field text* is limited to 140 characters while Twitter have added a new feature that allows users to extend the size of the text into another field called *extended_tweet*.

```
{
  "_id": "5c0ebdb444acc32008ac6cc1",
  "created_at": "Mon Dec 10 19:25:34 +0000 2018",
  "text": "@AM__ALHOSANI @M_Alyafeai لاعب نرجسي من يوم يومه",
  "in_reply_to_screen_name": "AM__ALHOSANI",
  "truncated": false,
  "in_reply_to_status_id": 1072178138904518700,
  "in_reply_to_user_id": 279595165,
  "favorited": false,
  "retweeted": false,
  "lang": "ar",
  .
. }
```

Figure 29. Example of one of the received objects from Twitter APIs

To determine which tweets should be delivered on streaming, a search string is passed to the streaming services. The streaming services will use the search string to filter and retrieve tweets that satisfy the search string. The search string is a list of comma-separated phrases. Each phrase may be one or more space-separated terms. The comma plays the role of logical (OR) operator while the space plays the role of logical (AND) operator. For example, "نادي الاهلي" means "نادي" AND "الاهلي" while "نادي، الهلال" means "نادي" OR

"الهلال". For AND operator, the order and the place of the terms were ignored by the search engine. For more information about the tips of the search string, please refer to Twitter documentation [118]. Figure 30 presents our search string that has been used to collect our dataset.

هر ذلك، هرد لك، نادي فقر، نادي أحد، نادي الوحدة، نادي الهلال، نادي النصر، نادي القادسيه،
نادي القادسية، نادي الفيصلي، نادي الفيحاء، نادي الفيحاء، نادي الفقر، نادي الفتح، نادي الشباب، نادي الرائد،
نادي الحزم، نادي التعاون، نادي الباطن، نادي الأهلي، نادي الأتفاق، نادي الأتحاد، نادي الاهلي، نادي الأتفاق،
نادي الأتحاد، نادي احد، نادي فقر، نادي أحد، نادي الوحدة، نادي الهلال، نادي النصر، نادي
القادسيه، نادي القادسية، نادي الفيصلي، نادي الفيحاء، نادي الفيحاء، نادي الفقر، نادي الفتح، نادي الشباب، نادي
الرائد، نادي الحزم، نادي التعاون، نادي الباطن، نادي الأهلي، نادي الأتفاق، نادي الأتحاد، نادي الاهلي، نادي
الأتفاق، نادي الأتحاد، نادي احد، لاعب، كوره، كورة، فقراوي، فريق فقر، فريق أحد، فريق الوحدة، فريق
الوحدة، فريق الهلال، فريق النصر، فريق القادسيه، فريق القادسية، فريق الفيصلي، فريق الفيحاء، فريق
فريق الفقر، فريق الفتح، فريق الشباب، فريق الرائد، فريق الحزم، فريق التعاون، فريق الباطن، فريق الأهلي،
فريق الأتفاق، فريق الأتحاد، فريق الاهلي، فريق الأتفاق، فريق الأتحاد، فريق احد، طحالب، طاقية، ضفدع،
زعيق، حكم المباره، حكم المبارة، بالتوفيق فرق، النادي، النادي، اللاعب، الفقراوية، الطواقي، الطحالب،
الضفادع، الأنديه، الأندية، الانديه، الأندية

Figure 30. Search string applied on Twitter APIs

The result of streaming is summarized in Table 25. We collected more than 3.5M tweets. After removing duplicate tweets, we ended up with around 919K tweets. We removed duplicates using Group-by operator in Mongoddb. The Group-by operator is applied on cleaned and normalized text.

Table 25. Abstract summary about the collected data

Factor	Statistics
Number of users by screen name	596525 users
Number of users by name	474774 users
Start date of tweets streaming	October 22, 2018 14:42:43
End date of tweets streaming	December 1, 2018 16:06:27
Number of tweets before removing duplicated	3529305 tweets
Number of tweets after removing duplicated	919483 tweets
Number of tweets contain emojis	248452 tweets out of 919483 tweets

Figure 31 presents more details about the number of tweets obtained per day and an abstract overview of time distribution.

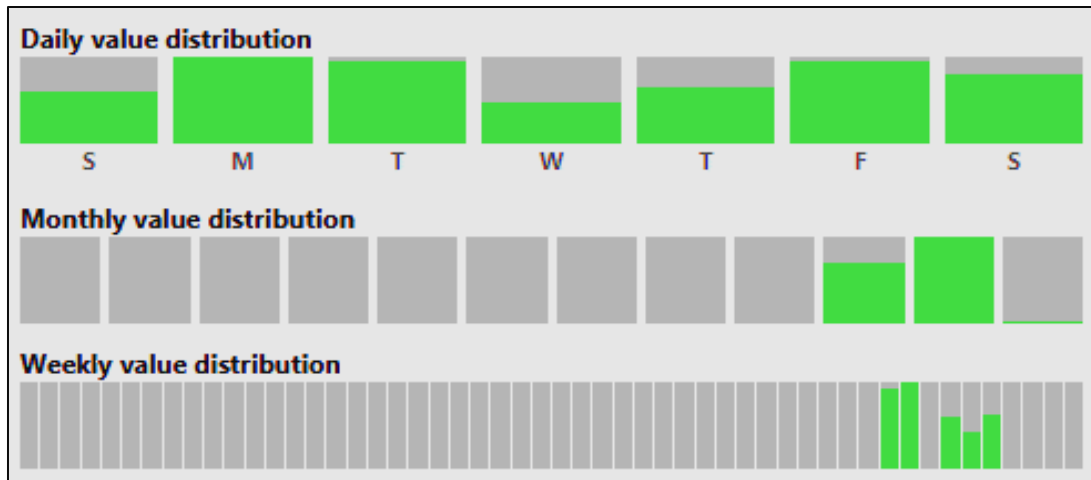


Figure 31. Time distribution of collected data

In Figure 32, we visualize the frequencies of the bigrams terms that have appeared in our search string. We can see that the term (نادي الاتحاد) has the highest frequency.

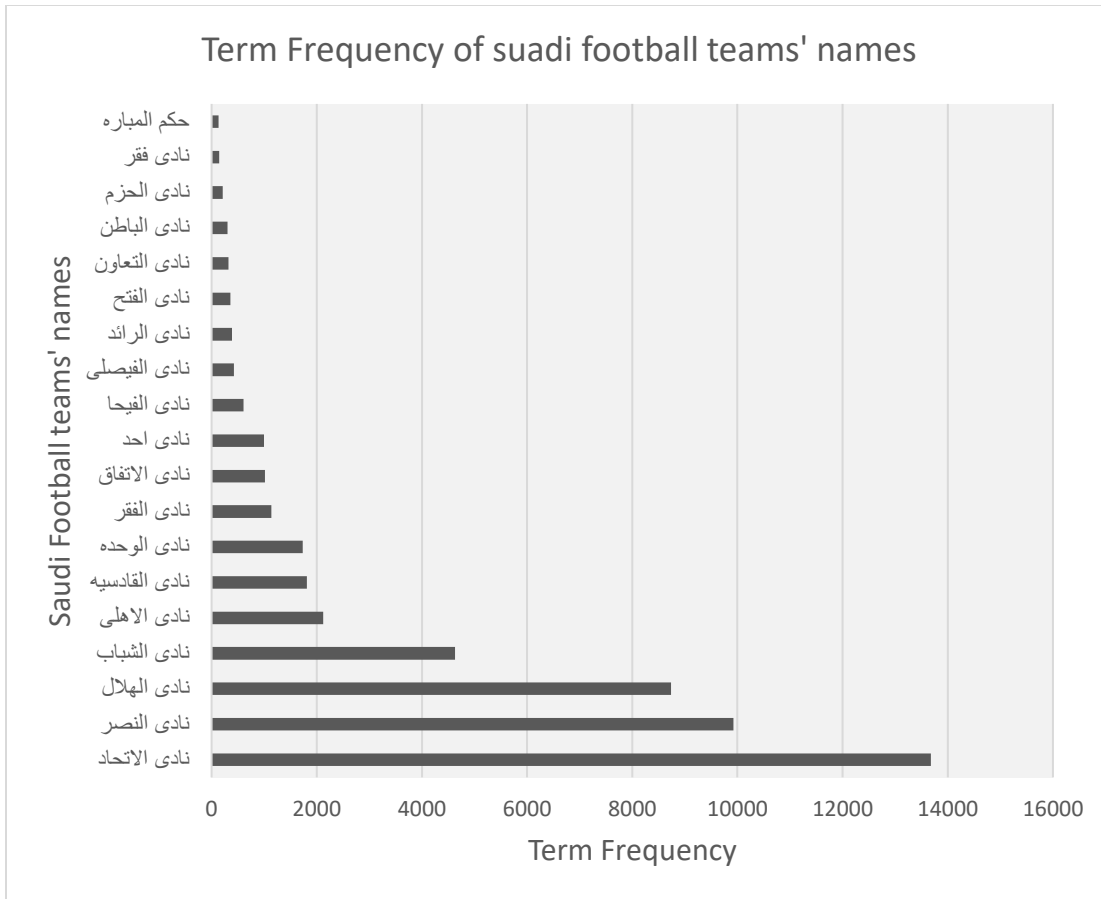


Figure 32. Frequencies of bigrams terms of the search string

In Figure 33, we visualize the frequencies of unigrams terms that had appeared in our search string. We can see that the general words were dominated.

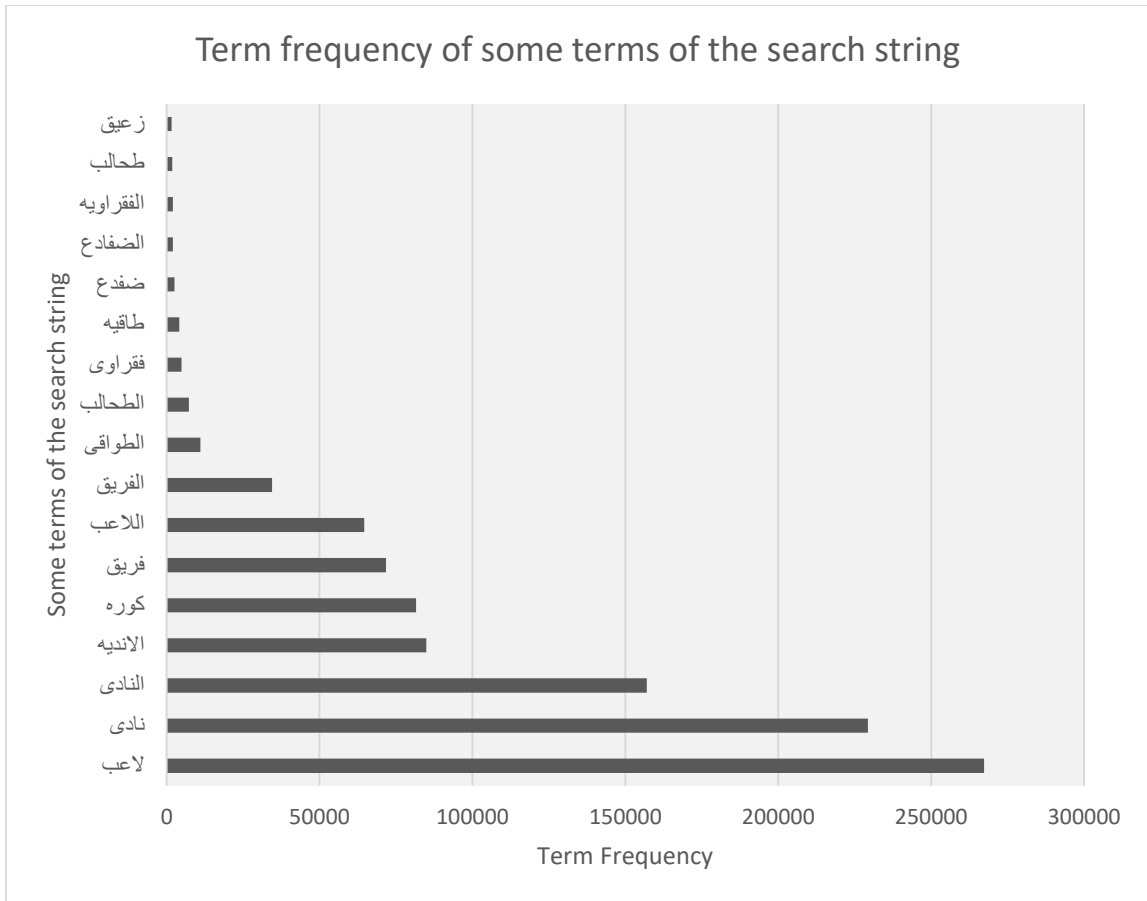


Figure 33. Unigrams term frequencies of the search string

In Figure 34, we visualize the frequencies of Saudi football names. We can see that the teams (الهلال، الاتحاد، الأهلي، النصر) are more popular. This conclusion may be biased to the short time where we spent around 40 days to collect the data. The collected data was in the beginning of the Saudi professional league which was started in August 30, 2018 and continued until middle of 2019.

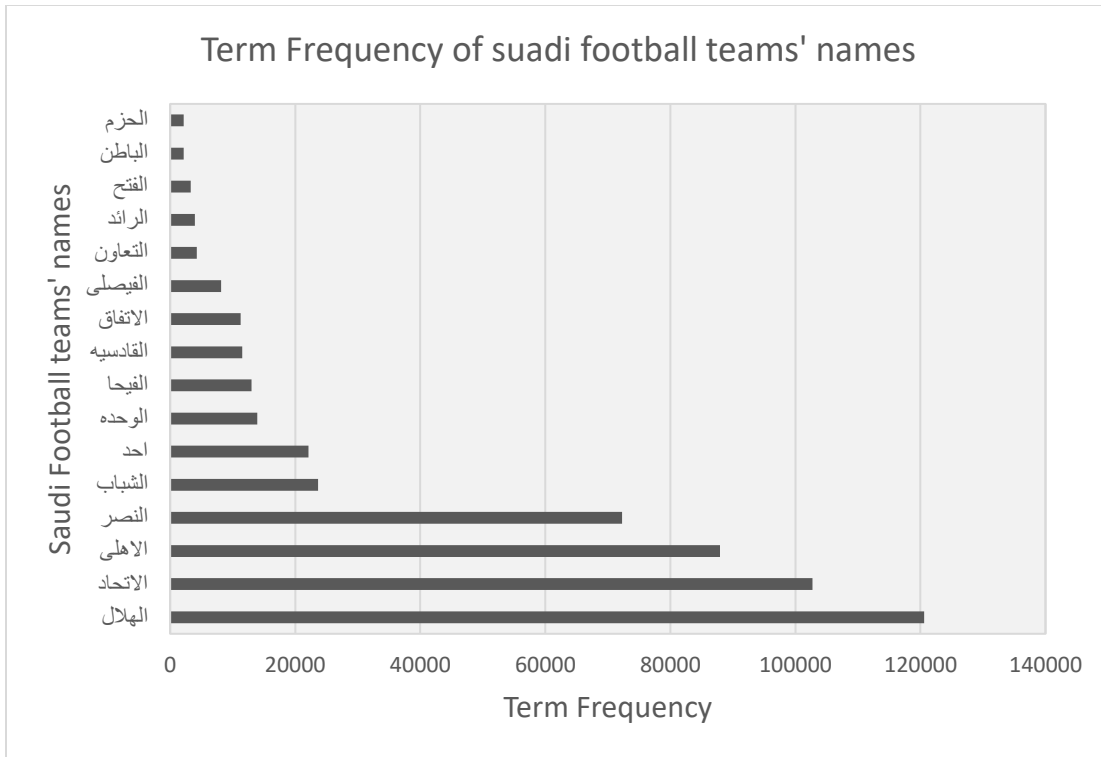


Figure 34. Frequencies of Saudi football teams' in the collected dataset

To extract some useful statistics about tweets' text and tweets' emojis, we have combined all tweets' text into one document. Then, we extract the number of words, emojis, characters, etc. Table 26 shows these statistics.

Table 26. Dataset statistics summary

Ngrams	Frequency
words	18561647
Emojis	782351
Characters	102137628
Unique words	481006
Unique emojis	2032
Unique characters	117

To have a more general idea about the collected data set, we have performed statistics for word unigrams, bigrams, and trigrams. We have also performed statistics on characters and emojis n-grams. The following tables presents the highest frequency terms for words, characters, and emojis. Table 27 presents the highest words' unigrams frequencies. Table 28 presents the highest words' bigrams frequencies. Table 29 presents the highest words' trigrams frequencies. Table 30 lists the highest frequencies of characters' unigrams. Table 31 shows the highest frequencies of emojis unigrams. Table 32 shows the highest frequencies of emojis bigrams. Table 33 shows the highest frequencies of emojis trigrams. these statistics are needed to help understanding the collected dataset.

Table 27. Frequencies of words' unigrams

فى	365891	من	363750	لاعب	267293	نادى	229296
على	190296	النادى	156976	و	144638	الهلال	120613
ان	113270	الاتحاد	102729	مع	100805	ما	88977
الاهلى	87936	الانديه	84950	الله	82433	كوره	81547
كل	80947	اللى	80464	لا	78585	هذا	73739
النصر	72301	فريق	71770	الى	69608	ولا	65388
اللاعب	64666	عن	63922	بس	62100	كان	60352

Table 28. Frequencies of words' bigrams

افضل لاعب	25304	لاعب فى	19321	النادى الاهلى	13891
نادى الاتحاد	13677	اكثر من	12021	اى لاعب	10319
فى النادى	10068	رئيس نادى	10044	فى نادى	10028
نادى النصر	9922	ربال مدريد	8913	شا الله	8854
نادى الهلال	8735	فى الدورى	7933	ال الشيخ	7485
فى كل	7350	كره القدم	7176	من النادى	7044
كل الانديه	7021	رئيس النادى	6770	فى العالم	6734
من نادى	6653	الاتحاد السعودى	6580	ان شا	6408
لاعب كبير	6355	الهلال الاتحاد	5883	لاعب من	5868

Table 29. Frequencies of words' trigrams

افضل لاعب فى	8455	ان شا الله	6256	تركى ال الشيخ	4805
محمد بن سلمان	3899	الامير محمد بن	2501	الله ونعم الوكيل	2414
لاعب فى العالم	2390	لاعب كره قدم	1613	احسن لاعب فى	1545
وان شا الله	1467	جائزه افضل لاعب	1463	هروب الاتحاد من	1460
نادى ليستر سيتى	1459	الهلال النفط	1453	حسبى الله ونعم	1376
		العراقى			
دورى ابطال افريقيا	1373	لاعب فى تاريخ	1373	فى كاس العالم	1284
افريقيا يا اهلى	1273	مجلس اداره نادى	1222	دورى ابطال اسيا	1183
دورى ابطال اوروبا	1181	النصر من الدورى	1161	فى الدورى السعودى	1148
الاتحاد من الاهلى	1147	انسحاب النصر من	1146	من الدورى مطلب	1140

Table 30. Frequencies of characters' unigrams

18382046	ا	14450580	ل	9747724	ي	4959412	م	4956419
و	ه	4519525	ن	4486757	ر	3971865	ب	3694924
ت	ع	3045757	د	3026314	ى	2898405	ف	2175249
ك	س	1873687	ق	1651875	ح	1585730	ج	1090684
ش	ص	772137	ط	730692	خ	688673	ز	518558
ض	ذ	411054	ث	318761	غ	311602	ئ	216743
180941	ظ	134099	و	44938	ء	16599	ف	2381
گ	چ	883	ى	866	أ	625	ه	370
ة	پ	277	ه	254	ك	242	ك	145
ل	ر	121	و	114	ن	93	ل	89
ژ	و	68	ي	65	ق	61		57
ے	و	44	ع	43	أ	42	ي	40
ر	ر	37	ه	35	ب	32	چ	32
ر	ء	31	ه	29	ن	20	ن	18
ر	ث	16	و	16	ء	11	ے	9
ن	چ	7	و	7	ء	7	ئ	7
آ	ئى	6	ا	5	ئى	5	ش	5
گا	ى	5	ب	4	ئى	4	ا	3
0	1	3	ن	3	ه	3	2	2
3	4	2	خ	2	ه	2	گا	2
ن	و	2	و	2	ه	1	5	1
6	7	1	ا	1	ن	1	ث	1
پى	ق	1	ف	1	ن	1	ل	1
ن	ه	1						

Table 31. Frequencies of emojis' and related-symbols' unigrams

	160799		46412		35520		26250
	21177		21066		19730		16868
	13750		13119		10660		8896
	8430		8259		7829		7755
	7489		6098		6053		5728
	5697		5668		5583	A	5563
	5242		5192		4452	♥	4296
	3749		3702		3658	S	3601
	3506		3484		3403		3354
	3197		3192		2689		2593
	2592		2260		2219		2101
	2100		2072		2065		2053
	2023	—	1934				

Table 32. Frequencies of emojis' and related-symbols' bigrams

	110230		16733		15246		12801
	7328		6242		5818		5704
	5540		4815		4701	••	4494
	4340		4340		4298		4160
	4035		3875		3748		3326
	3057	S A	3006		2909		2797
	2747		2704		2593		2515
	2392		2343		2250		2244
	2202		2118		2053		2038
	2008		1861		1822		1820
	1786		1753		1708		1688
	1669		1657		1579		1567
	1465		1465		1444		1365

Table 33. Frequencies of emojis' and related-symbols' trigrams

	78957		9383		8510		4591
	2868		2780		2674		2310
	2014		1993		1913		1626
	1556		1466		1464		1453
	1397				1370		1334
	1232				1167		1165
			1198				

We have developed some tools and resources to help identifying some elements of the dataset such as emojis. More information about the developed resources discussed in Chapter 7. The collected dataset is used to build the sentiment analysis resources such as annotated corpora, annotated senti-lexicons, general-background language model, etc. All these resources discussed in detail in Chapter 7.

CHAPTER 6

DATA PREPROCESSING

In this chapter, we present various performed preprocessing tasks. These tasks include cleaning, normalization, removing duplicate, light-stemming, lemmatization, segmentations, emojis extraction, and stop-words removal. The output of each task was added to a new field in each tweet. In other words, our dataset contains one attribute for each task where it has an attribute for cleaned text, an attribute for normalized text, an attribute for the output of the stemming, an attribute for the output of the lemmatization, an attribute for emojis, etc. Figure 35 shows one entity (one tweet object) of our preprocessed dataset.

```
{
  "_id": "5c40d94e71126412cc360a26",
  "text": "نادي الشعب [١٩٢٧]. تواجدمك يشرفنا#  
غير اتحادي لاتجي ❌👍. https://t.co/bdbGHWVkaA  
#تركي_اهبط_الاتحاد15",
  "cleaned": "نادي الشعب تواجدمك يشرفنا غير اتحادي لاتجي تركي اهبط الاتحاد",
  "normalized": "نادى الشعب تواجدمك يشرفنا غير اتحادي لاتجي تركي اهبط الاتحاد",
  "segments": "نادى ال شعب تواجدمك يشرفنا غير اتحادي ل اتجي تركي اهبط ال اتحاد",
  "prefixes": "ال ل ال",
  "suffixes": "كم نا",
  "light-stems": "نادى شعب تواجدمك يشرفنا غير اتحادي لاتجي تركي اهبط الاتحاد",
  "lemma": "نادى شعب تواجدمك يشرفنا غير اتحادي لاتجي تركي اهبط الاتحاد",
  "emojis": " B N 🧡🤔👍❌👍",
  .
}
```

Figure 35. A preprocessed example

analysis. For example, the words (أفضل) and (افضل) have the same meaning. Both are actually representing the same word but one of them is misspelled. This is common in social sites where some people write Alef with Hamza (ء) and others write Alef without Hamza. However, without normalization, a machine deals with both forms of the same word as they are different words where each word will be represented with separate feature in the feature vector space. The machine learning model can take a bag-of-words feature or other similar feature representation as an input. Removing the Hamza (ء) from (أفضل) can help to reduce the features' vector space. Another example, Table 35 and Table 36 illustrate how documents or tweets are represented in the feature vector space and how normalization helps to reduce the number of features. Therefore, we need to utilize, develop or adjust a normalizer to help us doing sentiment analysis. We have utilized and modified IBM normalizer [119] to achieve the needed normalization tasks including removing Arabic diacritics (tashkeel), removing tatweel, normalizing shapes of Alef, normalizing tah marbouta to hah, normalizing yah maksoura to alef maksoura, normalizing ben, and normalizing Abdal. Table 37 shows examples for these types of normalization.

Table 35 and Table 36 show examples of four annotated documents where three of them are considered as positive (non-fanatic) and the last one is considered as negative (fanatic). Both tables show how the raw data is converted to a feature vector space. The way of converting the string to feature vector is called vectorization or featurization. Table 35 shows how the non-normalized text is converted to feature vectors. Although, both words (أشقائنا) and (اشقائنا) have the same meaning (our brothers), they were represented by two different features. This non-normalized text increases the vector space and causes a negative impact when we conduct classification and machine learning. On the other hand,

Table 36 shows how normalized version is vectorized where both (أشقاءنا) and (اشقائنا) are represented as one feature vector.

Table 35. The Effect of non-normalized tweets on the size of features


(: + -:)	Tweets of Non-Normalized Form	Convert documents to feature vector space	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
			(أشقاءنا)	(اشقائنا)	(اشقاء)	(الشقاق)	(النفاق)	
+	شكراً <u>أشقاءنا</u> الهالين		1	0	0	0	0	...
+	<u>اشقائنا</u> في فريق الأهلي		0	1	0	0	0	..
+	جميعنا <u>اشقاء</u>		0	0	1	0	0	
-	فريق <u>الشقاق</u> والنفاق		0	0	0	1	1	

Table 36. The Effect of normalized tweets on the size of features


(: + -:)	Tweets of Normalized Form	Convert documents to feature vector space		Feature 2	Feature 3	Feature 4	Feature 5
			(اشقائنا)	(اشقاء)	(الشقاق)	(النفاق)		
+	شكراً <u>اشقائنا</u> الهالين			1	0	0	0	...
+	<u>اشقائنا</u> في فريق الاهلي			1	0	0	0	..
+	جميعنا <u>اشقاء</u>			0	1	0	0	
-	فريق <u>الشقاق</u> والنفاق			0	0	1	1	

Table 37. Data normalization examples

Task	Input	Output
Removing Arabic diacritics (tashkeel)	يُشار إلى أن اللُّغَةَ العَرَبِيَّةَ	يشار إلى أن اللغة العربية
Removing tatweel	مبـــــــــــــــروك	مبروك
Normalize alef	نادي الأهلي إلى النهائي (أآ)	الاهلي الى النهائي (III)
Normalizing teh marbuta to heh	ة	ه
Normalizing yeh maksura to alef maksura	"ي■"	"ى■"
Normalizing Ben	"■ ابن ■"	"■ بن ■"
Normalizing Abdal	"■ عبد ■ ال"	"■ عبدال■"
<p>Note: A rectangle black box before/ after some words represents a space.</p>		

6.3 Stemming and lemmatization

The normalized text could still be considered as under-preprocessing where there are some words need to be preprocessed more to reduce the feature vector space. For example, both words in Table 36 (اشقائنا) and (اشقاء) share the same meaning (brothers), and they should have one feature vector. Therefore, we need another type of preprocessing in order to unify words that have the same root or stem. Such operation is a kind of morphological analysis. Some morphological tasks were also important to reduce the dimensionality and improve the performance. Morphological analysis should be applied in sentiment analysis with caution as it may lead to over-preprocessing or under-preprocessing. The over-preprocessing may have negative impact on sentiment analysis where positive and negative words are reduced to one form that could lead to overlapping. For example, the words

(أشقائنا) and (الشقاق) can share the same root-stem (شق) while the first one is a positive word that means (our brothers) and the second one is a negative word that means (discord). If we conduct root-stemming to the dataset, both words will be represented in the feature space as one feature. Although the root-stemming can reduce the number of features dramatically, some words can be overlapped where the positive and negative words cannot be distinguished. For this, root-stemming is not recommended in Arabic sentiment analysis. On the other hand, under-preprocessing might lead to a high number of features where words with the same meanings could be represented in many features in the feature vector-space. For example, (أشقائنا) and (أشقاء) have the same meaning and should be represented as one feature in the feature vector-space. If we conduct segmentation and remove suffixes and prefixes of these two words, then we will have (أشقاى) and (أشقاء) where they are still represented as two different features and the segmentation cannot solve the problem. Therefore, we need to have a compromise to avoid under-preprocessing and over-preprocessing. While the segmentation might lead to under-preprocessing and the root-stemming might lead to over-preprocessing, the researchers in Arabic sentiment analysis recommend using something in between which is called light-stemming or lemmatization. Table 38 shows some examples of light-stemming analysis on some words. The best analysis level in these examples is the one that leads to two features. The table shows that segmentation will lead to 6 features when the prefixes and suffixes are considered as parts of the feature vector-space while extracting only the core-segment can lead to 4 features. In this case, segmentation can lead to under-preprocessing. On the other hand, the table shows that the root-stemming leads to one feature where it cannot distinguish between positive words and negative words. Lemmatization and light-stemming can help

in providing a compromise between under-preprocessing and over-preprocessing. We can see that lemmatization leads to 2 features which avoid the overlapping and reduce the number of features to acceptable level. Both lemmatization and light-stemming are recommended in Arabic sentiment analysis where they are used in many Arabic sentiment analysis publications such as the work of Refaee [82].

Table 38. Example of over-preprocessing and under-preprocessing

Given Words	Segmentation	Core Segment	Lemma	Light-Stemming	Root-Stemming	Polar	Note
أشقاؤنا	أشقاء+نا	أشقاء	شقيق	اشقاء	شق	positive	same words
أشقاء	أشقاء	أشقاء	شقيق	اشقاء	شق	positive	
الأشقاء	ال+أشقاء	أشقاء	شقيق	اشقاء	شق	positive	
أشقاؤكم	أشقاء+كم	أشقاء	شقيق	اشقاء	شق	positive	
شقيق	شقيق	شقيق	شقيق	شقيق	شق	positive	
الشقاق	ال+شقاق	شقاق	شقاق	شقاق	شق	negative	
#Features	6	4	2	3	1		
preprocessing	Under	under	Trad off	Trad off	over		

There are four levels of morphological analysis: segmentation, light-stemming, lemmatization and root-stemming. Segmentation helps to split the word into segments: prefixes, core-segment and suffixes. Therefore, core-segment is our target. Using core-segments as features helps to reduce the feature vector space. For example, both words (اشقاؤنا) and (اشقاؤكم) have the same meaning and will be segmented as (اشقاء+نا) and (اشقاء+كم). The core-segment of both words will be (اشقاء). Therefore, segmentation will help to represent these two words as one feature and reduces the feature vector space. Vectorizing segmented text can logically be considered better than vectorizing normalized text from the dimensionality reduction viewpoint. However, there are some cases that core-segment will not work. For example, both words (اشقاؤنا) and (الاشقاء) shared the same

meaning (brothers), and the segmentation cannot detect that these two words should be one feature where the core-segments (اشقائ) and (اشقاء) have different shapes and considered as two different features in the feature vector space. In similar cases, researchers move from segmentation to more precise analysis such as light-stemming or lemmatization. Light-stemming can perform segmentation with deeper analysis where it can do small adjustment to the end of the core-segment after removing suffixes and prefixes. For example, the word (الحضارات) can be segmented to (ال+حضر+ات) where the core-segment (حضر) is adjusted by the stemmer to be (حضارة). Another example is the word (اشقائنا) can be segmented to (اشقائ+نا) where the core-segment (اشقائ) is adjusted to be (اشقاء). As a result, the light-stemming can do what segmentation can do and figure out the cases that cannot be handled by segmentation to reduce the feature space. Although the light-stemming can do extra work to form the core-segment and reduce the feature space, it has some limitation to handle some cases that need modification on infixes to form the stem into more precise form to reduce the feature space. For example, both words (اشقائنا) and (شقيق) share the same meaning (brother). Light-stemming cannot figure out that these two words should have one feature in the feature input space. Light-stemming do not have the ability to deal with infixes and shape the two words into one form. Therefore, the lemmatization can help to do what light-stemming can with more ability to deal with infixes. In this case, lemmatization can transform both words into one feature (شقيق).

At this stage, the following question arises: What are the suitable morphological tools for Arabic sentiment analysis that help to conduct stemming, segmentation, lemmatization, POS Tagging, etc.? We have shown that lemmatization and light-stemming are recommended for Arabic sentiment analysis. Table 39 shows the performance of 5

morphological tools. We have examined: Farasa, Madamira, Assem's stemmer, NLTK, and Alkhalil2. Table 39 shows the differences among the four levels of the morphological analysis which are: segmentation, lemmatization, light-stemming and root-stemming. For lemmatization, we can notice that both Farasa and Madamira produce exactly the same results. From usability point of view, Farasa is preferable because it can be easily imported and accessed using java APIs. From the output point of view, the output of Farasa can be easily interpreted and understood. On the other hand, Madamira produces an xml file with more information about the analysis. Farasa output can be directly assigned to a java variable.

Table 39. Examining the impact of five stemmers

Tool	Farasa [109]	Farasa [109]	Madamira [120]	Assem (snowball) [121]	NLTK [122]	AlKhalil 2 [123] [124]	Human
Given words	Segmentation	lemmatization	lemmatization	Light-stemming	root stemming	root stemming	root stemming
أخوة	أخوة+ة	أخوة	أُخُوَّة	اخو	اخة	ءخو	اخ
جماهير	جماهير	جمهور	جُمهُور	جماهير	جماهير	جمهر	جمهر
الحبايب	ال+حبايب	حبايب	حبايب	حبايب	حبيب	حب	حب
خونة	خون+ة	خون	خَائِن	خون	خون	خون	خان+خون
رجال	رجال	رجل	رَجُل	رجال	رجل	رجل	رجل
بالأنسجام	ب+ال+أنسجام	أنسجام	أُنْسِجَام	انسجام	سجم	سجم	سجم
مستشفيات	مستشفيات+ات	مستشفى	مُسْتَشْفَى	مستشفيا	شفي	شفي	شفي
أشقائكم	أشقائكم	شقيق	شَقِيق	اشقاء	شقى	شقق	شقى
شقاق	شقاق	شقاق	شقاق	شقاق	شقق	شقق	شقى
أشقاء	أشقاء	شقيق	شقيق	أشقاء	شقىء	شقق	شقى
شقى	شقى	شقى	شقى	شقى	شقى	شقق	شقى
يتصل	يتصل	اتصل	أَتَّصَلَ	يتصل	تصل	صلى+ وصل	صل
المواصلات	ال+مواصل+ات	مواصل	مُواصلَة	مواصل	وصل	وصل	صل
التواصل	ال+تواصل	تواصل	تواصل	تواصل	وصل	وصل	صل
مصنوع	مصنوع	مصنوع	مَصْنُوع	مصنوع	صنع	صنع	صنع
أعزائنا	أعزائنا	عزيز	عَرِيز	اعزاء	عزئ	عزو	عز
عزيز	عزيز	عزيز	عَرِيز	عزيز	عزز	عزز	عز
أعزاء	أعزاء	عزيز	عَرِيز	اعزاء	عزه	عزز	عز

Tool	Farasa [109]	Farasa [109]	Madamira [120]	Assem (snowball) [121]	NLTK [122]	AlKhalil 2 [123] [124]	Human
يقول	يقول	قال	قال	يقول	يقول	قول	قل
وستكتشفون	و+س+تكتشف+ون	اكتشف	اكتشف	تكتشف	كشف	كشف	كشف
لاعبين	لاعب+ين	لاعب	لاعب	لاعب	لعب	لعب	لعب
أمرأة	أمرأة	أمرأة	أمرأة	امرء	مرأ	مرء	مرء
بنجاسة	ب+نجاس+ة	نجس	نجس	نجاس	نجس	نجس	نجس
تواجدكم	تواجد+كم	تواجد	تواجد	تواجد	وجد	وجد	وجد

To compare available stemmers, we have selected eight stemmers that can be easily imported and accessed using java APIs. In this comparison, we have used our lexicon-unigrams-terms as a dataset where the dataset contains 480 Arabic unigrams words. The dataset was manually prepared by removing possible prefixes and suffixes with some adjustments. Then, we compare the results of the stemmers with the results of the manual stemming. Table 40 shows the results of this comparison where Farasa dominates others with accuracy 68 %.

Table 40: Comparing eight stemmers

Stemmers	Number of Correct	Number of wrong	accuracy
Ara-Morph	129	351	0.268
Farasa	329	151	0.684
Shereen Khoja	129	351	0.268
Light10	270	210	0.562
Motazaad Lucene	271	209	0.564
Snowball Assem	249	231	0.518
Sandford Segmentation	269	211	0.560
Voting created by us	280	200	0.583

CHAPTER 7

BUILDING SENTIMENT RESOURCES

In this chapter, we present two types of fanatic-lexicons and two annotated fanatic-corpora. For the lexicons, one lexicon was manually built, and two lexicons were generated automatically. One of them was generated using Pointwise mutual information (PMI), and the other lexicon was constructed using the proposed Term Frequency – Inverse Context Frequency (TFICF). For corpora, the first corpus contains 276176 labeled tweets where the tweets were automatically annotated. The second corpus contains 11349 labeled tweets where the tweets were semi-automatically annotated. The process of creating these resources are presented in the following sections.

7.1 Building Fanatic-Lexicon Manually

In this section, the construction method of a manually fanatic-lexicon is described. We will consider this lexicon as the *core fanatic-lexicon*. This *core fanatic-lexicon* is one of our contributions to the research community. It contains 1766 unique phrases. The lexicon’s phrases are distributed in 21 contexts. These contexts were described in Chapter 4.

To build the *core fanatic-lexicon*, manual and semi-automatically approaches were followed. We have used “*sports-fanaticism in text*” formalism, proposed in chapter 4, as a guide to build this lexicon. Initially, we have suggested a keyword for each context in which the suggested keyword plays a vital role on the meaning of the context. Table 41 shows the initial keywords for each context.

Table 41: Keyword for each context

Positive & Negative Indicators	Context	Initial Keyword
Aggression	Threat	ادعسهم
	Insult	يلعن
	Revenge/Vengeance	سننتقم
Agitation	Accusation	منافق
	Promoting malpractices	أكسر
	No respect	لانحترم
Hatred	Hatred	كریه
	Contempt	طواقي
	Obduracy	عناد
Passion	Extreme grief	قهر
	Prejudice	الزعيم
	Indignation	جعل يهبط
Adaption	Sportsmanship	هاردلك
	Containment	حبايب
	Responsibility	شدو حليكم
Tolerance	Tolerance	آسف
	Cooperation	تعاون
	Flexibility	ريما
Respecting	Respect	احترامي
Broad minded	Knowledge	المنصوص
	Ideas visions	مؤشرات

To increase the number of the suggested keywords, we have utilized search and advance-search options available in Twitter [125] to find new related-words to each context's word. The way of finding new words is by words co-occurrence where new words can be extracted by analyzing the retrieved tweets. In details, each time, one of the keywords (presented in Table 41) along with some domain-related keywords (presented in Table 42), will be posted in the search box to retrieve related-tweets. Domain keywords are posted in the box called: *(Any of these words)*. Also, one of our research's context-related words is posted in the box called: *(This exact phrase)*. For example, Figure 36 shows how the advance search engine creates a search string that will be used to retrieve related tweets. We passed the word (هاردلك) as a context-related word along with some domain-related words (الهلال، النصر، الأهلي) (See Figure 36).

Table 42: Sport domain keywords

نادي الوحدة	نادي الوحدة	حكم المباراة	اللاعب	فريق	نادي
نادي الفيحاء	نادي الفيصلي	نادي القادسية	نادي القادسية	نادي النصر	نادي الهلال
نادي التعاون	نادي الحزم	نادي الرائد	نادي الشباب	نادي الفتح	نادي الفيحاء
	نادي أحد	نادي الاتحاد	نادي الاتفاق	نادي الاهلي	نادي الباطن

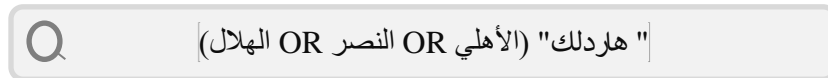


Figure 36. Twitter's search box

One

observation related to Twitter's search engine is that the search engine does some preprocessing tasks to normalize a given text. This is good for us as it limits the number of

possible search keywords. For example, if the word (الأهلي) with (ء) is passed to the search engine, then the search engine will retrieve also tweets that have the word (الاهلي) without (ء). After that, the retrieved tweets were analyzed to find and extract new context-related phrases where the new phrases were classified to the same context of the search keyword. This process was repeated to accumulate more context-related phrases.

Some context-related phrases were extracted from documents. These documents were downloaded from Saudi Arabian Football Federation (SAFF) website [126] and Almaany website [127]. The downloaded documents had been tokenized to extract unigrams, bigrams and trigrams. The output of tokenization is the ngrams terms and their frequencies. After analyzing the results, a threshold was used where only the high frequent ngrams will be analyzed and classified into context-related words. Why do we need to extract some terms from those documents? Those documents contain terms that are related to some contexts in our formalism such as the context of respecting rules and regulations. We have tried to extract some words that have high probability to be used in the context of respecting regulations such as (تنص اللائحة), (المنصوص), (المادة رقم), etc. Talking about such words may mean that the writers are aware about such regulations which might be considered as a kind of respecting them.

For semi-automation, Almaany dictionary was utilized to acquire the synonyms and antonyms of each keyword. The collected synonyms were stored in the database. Then, we have built a web application that simplifies the manually selection of related words and ignore others. Figure 37 shows a screenshot of this application.

Note	Context	Term	Ngram	Core1	All Forms	Unrelated Forms
	accusation	حرامي	1	<input type="checkbox"/>	27	16
	accusation	خانن	1	<input checked="" type="checkbox"/>	13	1
	accusation	خونة	1	<input checked="" type="checkbox"/>	35	18
	accusation	سارق	1	<input checked="" type="checkbox"/>	23	7
	accusation	سرق	1	<input checked="" type="checkbox"/>	141	8
	accusation	صهاينة	1	<input type="checkbox"/>	5	0
	accusation	صهيووني	1	<input type="checkbox"/>	5	0
	accusation	فاسد	1	<input checked="" type="checkbox"/>	40	3
	accusation	فاشل	1	<input checked="" type="checkbox"/>	46	0
	accusation	لصوص	1	<input type="checkbox"/>	10	2

Figure 37. Web app to simplify adding new context-related words and expressions

As a result, a *manual fanatic-lexicon* has been created with 1864 phrases (1766 unique phrases) distributed in 21 contexts. Some phrases appeared in more than one context within the same orientation (non-fanatic or fanatic). While the *core fanatic-lexicon* has limited number of entities (1766 unique phrases), it has been used as seed-keywords to automatically extract large-scale lexicon from large corpus. In the next section, we will discuss our approach of constructing large-scale fanatic-lexicons.

7.2 Building Fanatic-Lexicon Automatically

In the previous section, a manual fanatic-lexicon was constructed with 1766 unique words distributed in 21 contexts. The manually constructed lexicon is limited to small number of phrases which may suffer from recall when applied to reality. From the literature, the manual generated lexicons are usually used as seed-words to generate large-scale lexicon automatically (See Aldayel and Haifa [35] and Mahyoub et al. [36]). There are three approaches to generate lexicons automatically: dictionary-based method, translation-based

method, and corpus-based method as discussed in the literature (see Section 2.2.1). In the previous section, we followed shallow dictionary-based approach to extend the manually created lexicon using Almaany dictionary [128]. However, the dictionary-based method and the translation-based method are limited to MSA as discussed in Section 2.2. While we need to generate fanatic-lexicons for social media text which is rich of dialects, the corpus-based approach is our target. From the literature, there are several methods to automatically generate a senti-lexicon from a corpus. Clearly, a common statistical-based method to generate senti-lexicon form corpus is Pointwise mutual information (PMI) [39] [37] [57]. PMI is adapted in this work to generate large-scale fanatic-lexicon. PMI is a statistical association used to measure the association strength between a term and the positive and the negative categories (See Al-Twairesh [57]).

7.2.1 Generate Lexicon Using PPMI

Pointwise mutual information (PMI) between two events, x and y , is the probability that the two events are occurring together, divided by the product of the probability of the two events occurring independently. It is answering the question: Do event x and y co-occur more often than if they are independent? PPMI is stands for Positive PMI where all the PMI values less than zero are replaced with zero (See Niwa and Nitta [129]). The following details describe how we can use PPMI method to generate multi-sentiment lexicons.

Method 1.0: Positive Pointwise Mutual Information (PPMI)

Input: M*N Word-Context matrix where rows in the matrix represent words $W = \{w_1, w_2, \dots, w_m\}$, columns represent contexts $C = \{c_1, c_2, \dots, c_n\}$, and each cell $f(w_i, c_j)$ represents the frequency of word w_i in context c_j .

Output: M*N Word-Context matrix where each cell $f(w_i, c_j)$ represents score of word w_i in context c_j where the score is calculated by PPMI.

1. Compute the joint probability of word w_i occur in context c_j

$$p(w_i, c_j) = \frac{f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)}$$

2. Compute the probability of word w_i using the following formula:

$$p(w_i) = \frac{\sum_{j=1}^C f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)}$$

3. Compute the probability of context c_j using the following formula:

$$p(c_j) = \frac{\sum_{i=1}^W f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)}$$

4. Compute pmi_{ij} using the following formula:

$$pmi_{ij} = \log_2 \frac{p(w_i, c_j)}{p(w_i)p(c_j)}$$

5. Compute $ppmi_{ij}$ using the following:

$$ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

As stated in Equation 1, If the two events occur more often together, then the numerator in Equation 1 will be much higher than the denominator.

$$pmi(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)} \quad (1)$$

As describe by Churck & Hanks [130], the PMI between two words X and Y answers the following question: Do words X and Y co-occur more than if they are independent? As described in 2

$$pmi(word_1, word_2) = \log_2 \frac{p(word_1, word_2)}{p(word_1) p(word_2)} \quad (2)$$

PMI between words and contexts can be computed as follows:

$$pmi(word, context) = \log_2 \frac{p(word, context)}{p(word) p(context)} \quad (3)$$

Positive PMI (PPMI) between two words can be computed by replaces all PMI values less than zero with zero. Before we adapt the PPMI to construct our lexicons, we need to represent some definitions as follows:

Definition 1: Our term-context matrix is defined as follows: if a matrix F with W row (words) and C columns (context), then $f(w_i, c_j)$ is the number of times word w_i occurs in context c_j . $f(w_i, c_j)$ refers to the frequency of the word w_i occurs in context c_j .

$$p(w_i, c_j) = \frac{f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)} \quad (4)$$

$p(w_i, c_j)$ is the joint probability of word w_i occur in context c_j where it is the frequency of a word w_i appears in the context c_j , normalized by the sum of all frequencies of all words in all contexts.

$$p(w_i) = \frac{\sum_{j=1}^C f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)} \quad (5)$$

$p(w_i)$ is the probability of word w_i which is the sum of all its frequencies in all contexts, normalized by the sum of all frequencies of all words in all contexts.

$$p(c_j) = \frac{\sum_{i=1}^W f(w_i, c_j)}{\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)} \quad (6)$$

$p(c_j)$ is the probability of context c_j which is the sum of all frequencies of all words appear in that context, normalized by the sum of all frequencies of all words in all contexts.

$$pmi_{ij} = \log_2 \frac{p(w_i, c_j)}{p(w_i)p(c_j)} \quad (7)$$

pmi_{ij} is the probability the word w_i and context c_j occur together over the product of the probability of that word and the probability of that context. Then, we take the log, and that's our PMI. As shown in

$$ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

positive PMI is extracted by replacing the values that are less than zero by zero. The following steps show how we have applied PPMI on our research.

Step 1: the collected corpus discussed in Chapter 5 is used. 919K unique tweets were annotated using our manual fanatic-lexicon. As a result, 300K tweets were annotated. The results of annotation have produced the eight indicators discussed in Chapter 4 where we called them here “contexts”. Table 43 shows an example where eight tweets are represented along with their annotation scores for each indicator/feature (context). See Section 7.3 for more details about the annotated corpus.

Table 43. Examples of Annotated dataset using the manual-fanatic-lexicon

Tweets	Fanatic (Negative)				Non-fanatic (Positive)			
	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
كل الاحترام لك 🤝❤️							1	
حزين جداً				1				
من جد شكراً						1		
فقراوي وضبع 😏😏			2					
شدوا الهمة شباب					1			
شف الجلد ي فقراوي 🙌👮	1		1					
طل سرقة بقا		1						
الخيبري محور ممتاز								1
.								
.								

Step 2: We have converted the annotated corpus with 300K tweets into term-context matrix. Rows in the matrix represent words W and columns represent contexts C . Each cell in the matrix refers to the frequency of the word w_i occurs in context c_j . Table 44 displays some examples of our term-context matrix for normalized text.

Table 44. An example of term-context matrix

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
خوى	12	7	32	31	7	14	168	7
دعس	405	6	137	63	6	26	38	18
ضفدع	155	71	1820	136	21	61	24	29
طواقي	19	21	382	52	15	16	4	11
فقرأوى	217	95	3657	443	48	129	41	94
فى	8611	11388	47660	49280	11661	32289	12356	22934
محترم	88	67	295	294	83	279	2199	78
من	10959	12004	60707	46120	12121	30776	13215	20235
هاردلك	28	32	157	209	690	162	49	65
				.				
				.				

Step 3: In this step we apply PMI. To compute PMI on our examples, we need to compute each term in the Equations 4, 5, 6, 7.

The denominator part of Equations 4, 5 and 6 ($\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)$) is computed by summing all frequencies in the whole matrix. For our example, we will only use those values presented in Table 44. We ignore values for remaining words that do not appear in this example. This sum of $\sum_{i=1}^W \sum_{j=1}^C f(w_i, c_j)$ results in 415629 frequencies as a total sum of all frequencies in the matrix. This part $\sum_{j=1}^C f(w_i, c_j)$ is computed by summing row-wise values in the matrix. It sums all frequencies in all context for each word w_i . A single vector (column) is produced where each value refers to a total sum of a word in all contexts. While the example contains 9 words, the computed vector will have 9 values: one for each word as shown in the column at the right side of Table 45.

This part $\sum_{i=1}^W f(w_i, c_j)$ can be computed by summing column-wise values in the matrix. It sums all frequencies of all words for each context c_j . A single vector (row) is produced where each value refers to a total sum of frequencies of each context c_i for all words. While the example contains 8 contexts, the computed vector will have 8 values: one for each word as shown in the row below Table 45.

Table 45. Example for computing the PPMI equation's terms

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge	$\sum_{i=1}^c f(w_i, c_j)$
خوی	12	7	32	31	7	14	168	7	278
دعس	405	6	137	63	6	26	38	18	699
ضفدع	155	71	1820	136	21	61	24	29	2317
طواقی	19	21	382	52	15	16	4	11	520
فقراوی	217	95	3657	443	48	129	41	94	4724
فی	8611	11388	47660	49280	11661	32289	12356	22934	196179
محترم	88	67	295	294	83	279	2199	78	3383
من	10959	12004	60707	46120	12121	30776	13215	20235	206137
هاردلك	28	32	157	209	690	162	49	65	1392

$\sum_{i=1}^W f(w_i, c_j)$									$\sum_{i=1}^W \sum_{j=1}^c f(w_i, c_j)$
	20494	23691	114847	96628	24652	63752	28094	43471	

Now, we are ready to apply the Equations 4, 5, 6, 7. We show here how to find pmi_{11} for the word w_1 (خوی) occurred in context c_1 (Aggression). The remaining can be computed by repeating similar process.

$$p(w_1, c_1) = \frac{12}{415629} = 0.00003$$

$$p(w_1) = \frac{278}{415629} = 0.00067$$

$$p(c_1) = \frac{20494}{415629} = 0.049308$$

$$pmi_{11} = \log_2 \frac{0.00003}{0.00067 * 0.049308} = \log_2 0.875418 = -0.19196$$

This example shows that the PMI for the word (خوى) occurred in context (Aggression) is (-0.19196) which will be zero when we apply the PPMI in equation 8. Table 46 shows the output of applying PMI on entire matrix of our example. Table 47 shows the output when PPMI is applied.

Table 46. The output of applying (PMI) on the example of Table 44

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
خوى	-0.19	-1.18	-1.26	-1.06	-1.24	-1.61	3.16	-2.05
دعس	3.55	-2.73	-0.50	-1.37	-2.79	-2.04	-0.31	-2.02
ضفدع	0.44	-0.90	1.51	-1.99	-2.71	-2.54	-2.71	-3.06
طواقى	-0.43	-0.50	1.41	-1.22	-1.04	-2.32	-3.14	-2.31
فقراوى	-0.10	-1.50	1.49	-1.31	-2.55	-2.49	-2.96	-2.39
فى	-0.17	0.03	-0.19	0.11	0.00	0.10	-0.10	0.16
محترم	-0.92	-1.53	-1.66	-1.42	-1.27	-0.90	3.27	-2.18
من	0.11	0.03	0.09	-0.06	-0.01	-0.04	-0.08	-0.09
هاردلك	-1.29	-1.31	-1.29	-0.63	3.06	-0.40	-0.94	-1.16

Table 47. The output of applying (PPMI) on the example of Table 44

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
خوی	0	0	0	0	0	0	3.16	0
دعس	3.55	0	0	0	0	0	0	0
ضفدع	0.44	0	1.51	0	0	0	0	0
طواقی	0	0	1.41	0	0	0	0	0
فقرای	0	0	1.49	0	0	0	0	0
فی	0	0.03	0	0.11	0.003	0.10	0	0.16
محترم	0	0	0	0	0	0	3.27	0
من	0.11	0.03	0.09	0	0	0	0	0
هاردلك	0	0	0	0	3.06	0	0	0

Previously, one example was examined to show how we have adapted PPMI to generate the proposed lexicons. Pre-smoothing was also examined by adding one to each term frequency. One observation is that stopwords were penalized while the important words were rewarded. As a result, 7 fanatic-lexicons were generated using PPMI. Three of the generated lexicons are for emojis and related-symbols and four lexicons are generated from text as shown in Table 48 and Table 49.

Table 48. Generated fanatic-lexicon for ngrams text

Text Form	Ngrams	Size
stems	unigrams	24438
normalized	unigrams	59373
stems	bigrams	155004
stems	trigrams	80374

Table 49. Generated fanatic-lexicon for Emojis and related-symbol

Ngrams	Size
unigrams	692
bigrams	1714
trigrams	1384

7.2.2 Generate Lexicon Using the proposed-TFICF

TFIDF stands for Term Frequency – Inverse Document Frequency. TFIDF is used to show how important is a word to a document in a corpus. In our case, we have adapted TFIDF to show how important is a word to a context in a corpus. TFIDF is computed by multiplying TF with IDF. We have proposed heuristic adjustment to adapt TFIDF with a purpose of building lexicons. Because we deal with a context not a document, we called the adapted version as TFICF which stands for Term Frequency – Inverse Context Frequency. Two modifications have been added to make the conversion of the TFIDF to new form TFICF in order to make it applicable to be used as new method for generating multi-sentiment lexicons which is described as follows:

Method 2.0: Term-Frequency – Inverse-Context Frequency (TFICF)

Input: M*N Word-Context matrix where rows in the matrix represent words $W = \{w_1, w_2, \dots, w_m\}$, columns represent contexts $C = \{c_1, c_2, \dots, c_n\}$, and each cell $f(w_i, c_j)$ represents the frequency of word w_i in context c_j .

Output: M*N Word-Context matrix where each cell $f(w_i, c_j)$ represents score of word w_i in context c_j where the score is calculated by the proposed-TFICF.

1. Compute the *Term-Frequency (TF)* of each cell using the following formula.

$$TF(w_i, c_j) = \frac{f(w_i, c_j)}{\sum_{i=1}^W f(w_i, c_j)}$$

2. Normalize the computed *Term-Frequency (TF)* of each cell using the following formula:

$$normalizedTF(w_i, c_j) = \frac{TF(w_i, c_j)}{\sum_{j=1}^C TF(w_i, c_j)}$$

3. Compute *Inverse-Context Frequency (ICF)* of each word by computing the variance among all contexts C for each word w_i using the notation:

$ICF(w_i) = \text{the standard deviation (std) of a word } w_i \text{ in all context } C$

4. Computing *TF-ICF* using the following formula:

$$TFICF(w_i, c_j) = normalizedTF(w_i, c_j) * ICF(w_i)$$

Step 1: we have converted our annotated corpus into a term-context matrix as shown in Table 44. Then, term frequency is computed using equation 9 where $f(w_i, c_j)$ is the frequency of a word w_i in a context c_j . The denominator refers to the total sum of all frequencies in the context c_j .

Table 50 shows the result of applying equation 9 on the term-context matrix.

$$TF(w_i, c_j) = \frac{f(w_i, c_j)}{\sum_{i=1}^W f(w_i, c_j)} \quad (9)$$

$$TF(w_i, c_j) = \frac{\text{the frequency of a word } w_i \text{ in a context } c_j}{\text{the total frequency of all words in a context } c_j} \quad (10)$$

Table 50. Applying TF on the term-context matrix of Table 44

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
خوی	0.00002	0.00001	0.00001	0.00001	0.00001	0.00001	0.00027	0.00001
دعس	0.00074	0.00001	0.00005	0.00003	0.00001	0.00002	0.00006	0.00002
ضفدع	0.00028	0.00012	0.00063	0.00006	0.00004	0.00004	0.00004	0.00003
طواق	0.00003	0.00004	0.00013	0.00002	0.00003	0.00001	0.00001	0.00001
فقاوی	0.00040	0.00016	0.00126	0.00019	0.00008	0.00008	0.00006	0.00010
فی	0.01575	0.01953	0.01646	0.02102	0.02054	0.01980	0.01958	0.02555
محرّم	0.00016	0.00011	0.00010	0.00013	0.00015	0.00017	0.00349	0.00009
من	0.02004	0.02059	0.02096	0.01967	0.02135	0.01887	0.02095	0.02255
هاردلك	0.00005	0.00005	0.00005	0.00009	0.00122	0.00010	0.00008	0.00007
				.				

Step 2: Now, our term-context matrix is representing the extracted TF. In this step, we apply row-wise normalization where the extracted TF of a word w_i in a context c_j is normalized by the total sum of TF of a word w_i in all context. This normalization can be achieved using Equation 11. Then, the standard deviation (STD) is extracted from the normalized values. STD is used as IDF in TFIDF (see Equation 12). The idea of selecting STD as IDF is that if the variance of a word w_i is high among all contexts, then the word w_i is important word to specific context/contexts and should be rewarded. On the other

hand, if the variance of a word w_i is low among all contexts, then the word w_i is not important to a specific context/contexts and should be penalized. Therefore, the value of the standard deviation can be adapted as IDF.

Table 51 shows the result of row-wise normalization using Equation 11. The table also shows a single vector for standard deviation values.

$$normalizedTF(w_i, c_j) = \frac{TF(w_i, c_j)}{\sum_{j=1}^C TF(w_i, c_j)} \quad (11)$$

$$IDF(w_i) = \text{the standard deviation of a word } w_i \text{ in all context } C \quad (12)$$

Table 51. Row-wise normalization for (TF) values

Term	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge	STD As IDF
خوى	0.062	0.034	0.031	0.037	0.035	0.024	0.754	0.022	0.25
دعس	0.795	0.011	0.051	0.029	0.011	0.017	0.065	0.022	0.27
ضفدع	0.229	0.098	0.508	0.047	0.030	0.030	0.031	0.026	0.17
طواقي	0.124	0.129	0.472	0.079	0.094	0.035	0.023	0.044	0.15
فقراوى	0.169	0.069	0.539	0.081	0.036	0.034	0.028	0.045	0.17
في	0.100	0.123	0.104	0.133	0.130	0.125	0.124	0.162	0.02
محترم	0.037	0.026	0.023	0.029	0.033	0.039	0.793	0.020	0.27
من	0.121	0.125	0.127	0.119	0.129	0.114	0.127	0.137	0.01
هاردلك	0.030	0.032	0.032	0.052	0.709	0.058	0.045	0.042	0.24
				.					.
				.					.

After computing the proposed TF and ICF, The TFICF is calculated as follows:

$$TFICF(w_i, c_j) = normalizedTF(w_i, c_j) * ICF(w_i) \quad (13)$$

Table 52. display the results after applying TFICF. We have followed this approach to generate three emojis fanatic-lexicons and four text fanatic-lexicons. One observation was that the stop-words (e.g في) were penalized while the important words were emphasized (rewarded).

Table 52. The result of the proposed TFICF

Term	Fanatic (Negative)				Non-fanatic (Positive)			
	Aggression	Agitation	Hatred	Passion	Adaption	Tolerance	Respecting	Knowledge
خوى	0.016	0.009	0.008	0.010	0.009	0.006	0.192	0.006
دعس	0.216	0.003	0.014	0.008	0.003	0.005	0.018	0.006
ضفدع	0.039	0.017	0.086	0.008	0.005	0.005	0.005	0.004
طواقى	0.018	0.019	0.069	0.012	0.014	0.005	0.003	0.006
فقرأوى	0.029	0.012	0.093	0.014	0.006	0.006	0.005	0.008
فى	0.002	0.002	0.002	0.003	0.002	0.002	0.002	0.003
مأترم	0.010	0.007	0.006	0.008	0.009	0.011	0.214	0.005
من	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
هاردلك	0.007	0.008	0.007	0.012	0.167	0.014	0.011	0.010

In this section, two methods were applied to automatically generate fanatic-lexicons which are: PPMI and proposed-TFICF. Both methods will be evaluated and compared in Chapter 9.

7.3 Building Fanatic-corpus Automatically

In this section, we will show the annotation process for the collected corpus, discussed in Chapter 5. The corpus was automatically annotated using fanatic-keywords. From the literature, automatically annotation is usually done using seed-words (keywords), using emoticons, or using hashtags. Noticeably, automatically annotation helps to create large-

scale corpora. In this work, the *core fanatic-lexicon*, represented in Section 7.1, was used as keywords to label each tweet into its contexts. The *core fanatic-lexicon* contains 1766 unique phrases distributed over 21 contexts. Therefore, 1766 keywords were used to label (919K) tweets into their target orientation. While the collected tweets were stored in MongoDB database, the MongoDB's query-engine was utilized using Java APIs to query tweets using one keyword at a time and to label the retrieved tweets by the label of the used keyword. This process was repeated for all keywords. For example, if the tweet (شف الجلد ي فقرأوي) existed in the database, and both keywords (الجلد) and (فقرأوي) have passed to the query-processor one at a time, then this tweet could have two labels. Each label is represented as a separate attribute. While we have 21 contexts, the result of annotation produces 21 attributes for each tweet. The annotation shows the count of context-keywords existed in tweets. After that, the 21 contexts were aggregated to our 8 fanatic-indicators/features where these indicators/features were used to decide the target labeled (non-fanatic or fanatic), as represented in the formalism (See Chapter 4). The following Figure 38 shows an example of annotated tweet.


```

{
  "text": "شف الجلد والدعس ي فقراوي (فقرأوي)",
  "cleaned": "شف الجلد والدعس ي فقراوي",
  "normalized": "شف الجلد والدعس ي فقراوى",
  .
  .
  "context_thread": 2,
  "context_thread_reason": ['الدعس', 'الجلد'],
  "context_insult": 0,
  "context_insult_reason": [],
  "context_revenge": 0,
  "context_revenge_reason": [],
  .
  .
  "context_contempt": 1,
  "context_contempt_reason": ['فقراوى'],
  .
  "indicator_aggression": 2,
  "indicator_agitation": 0,
  "indicator_hatred": 1,
  .
  .
  "target_label": "fanatic",
  .}

```

Figure 38. An example for an annotated tweet

As a result of this process, 300K tweets were annotated according to the proposed formalism where the annotated tweets have information about 21 contexts and eight indicators/features. We have also included 21 reason attributes (one for each context). The reason attribute tells us why this tweet belongs to this context. To justify, the keyword K that was passed to the query-engine, is our reason to all retrieved tweets. For example, if a tweet has two keywords of the context (contempt) which are (طواقى) and (فقراوي), then, the attribute contempt will have score (2) and the attribute contempt-reason will contain these two words, (طواقى) and (فقراوي). As far as we know, the proposed corpus is the first corpus

for sports-fanaticism. It can be used as a foundation for other studies in fanatic-classification. The proposed corpus has many domain-related attributes which help to conduct deep analysis of sport-fanaticism.

7.4 Building Fanatic-corpus Semi-Automatically

To construct a representative test dataset, we have built a fanatic-corpus semi-automatically. We need our test data to satisfy the following conditions to be a good measure for our training dataset:

- *High coverage:* We need our test dataset to cover most of the cases required to measure the classification models.
- *Low redundancy:* We need to ensure that test dataset is not biased to specific observations, contexts, indicators and keywords in our formalism.
- *Balanced:* We need to ensure that the test dataset is not biased to specific orientation (non-fanatic or fanatic).

We have collected our test dataset from Twitter. The test data is not part of the 3.5M tweets corpus that have been used to build the resources (training dataset and lexicons). The test dataset was collected in a different time from the time of tweets used to build the corpus and the resources (training dataset and lexicons). It was collected between 09, Dec 2018 and 16, Dec 2018. We have collected around 300K tweets.

Removing duplicates have reduced the number of tweets to 53696 tweets. After that, we have performed automatic annotation using our *core fanatic-lexicon*. As a result, 23223

tweets were covered by the lexicon terms. This process added 21 context-related features and 21 context-reasons as illustrated in Figure 38. We have extracted other features for each tweet such as high-frequent word, number of words and number of emojis.

To reduce redundancy, we have applied group-by all extracted features: 21 context-related features, 21 context-reason features and other 3 features. This has eliminated redundant features from the perspective of domain-related contexts and phrases. As a result, 11349 tweets were retained to be part of our manual validation and annotation process. Table 53 provides information about this test dataset.

Table 53: Test dataset Statistics

Metrics	Statistics
Number of tweets	11349
Start date	Sunday, Dec 09, 2018, 20:18:54
End date	Sunday, Dec 16, 2018, 05:59:29
Number of users	9478
Number of locations	2655
String size	1851611
Average tweet's size	163.2
Number of positive (non-fanatic)	8659
Number of negative (fanatic)	2690

To conduct manual validation, we have built a web application for this purpose. We have prepared a video that shows the process of manual validation through the developed web-app. Copy of the video could be reached by visiting the link (<https://youtu.be/9CQaVTRA6D8>) [131]. The validation process has been done by the author of this thesis. The manual validation showed that 10808 tweets were annotated

correctly to their target orientations (non-fanatic or fanatic), while 749 tweets were wrongly annotated. Therefore, the annotator has inverted the polarity of the wrong 749 tweets to their correct orientation. The validation outputs are illustrated in Table 54 and Table 55.

Table 54. Evaluation results of the manual fanatic-lexicon

Tweets Status	#correct	#wrong	accuracy
Tweets that have strong evidence (more than one polar-phrases)	4418	143	96.86%
Tweets that have weak evidence (one polar-phrase)	4332	188	95.84%
Tweets that have mix polar-phrases (non-fanatic and fanatic phrases)	1850	418	81.57%
overall	10600	749	93.40%

Table 55. Contents comparison of the manual fanatic-lexicon evaluation

Tweets Status	#correct	#wrong	accuracy
with emojis	2847	242	92.17%
without emojis	7753	507	93.86%
overall	10600	749	93.40%

During validation, the annotator has found that there were some words in the lexicon that have word sense ambiguities (WSA). Examples of this issue are shown in Table 56.

Table 56. Examples of ambiguous Arabic terms

Non-fanatic	Fanatic	Non-fanatic Context	Fanatic Context
no matter بغض	hate بغض	هذه فرصة جيدة (بغض النظر) عمن يفوز	رئيس نادي يدافع عن نادي ثاني، ما هو محبة لكن (بغض) في الهلال
contract عقده	inferiority complex عقده	اللاعب جدد (عقده) مع النادي	هذا اللاعب عنده (عقدة) نقص
football كره	hate كره	من الممكن ان تصبح لاعب (كره) رائع	ترى ما هو حب في النصر وانما (كره) الهلال
feel نحس	jinx نحس	الهلال الآن (نحس) انه بدون رئيس	لاعب (نحس)، وفريق منحوس

The annotator also has observed that some of incorrect annotation came from the negation. Table 57 shows some examples of some negated words.

Table 57. Examples of incorrect annotations due to negation

بلا طموح
بلا هدف
بلاش نكسر
بدل من تحطيم

To generate a balanced test dataset, we have applied stratified sampling technique to conduct under-sampling, as our data have several categories and sub-categories. We have performed under-sampling for each category and sub-category. We have performed the following under-sampling techniques [74]:

- Random under-sampling
- Clustering under-sampling
- High-length Under-sampling
- Low-length under-sampling

By applying these techniques, we have produced four test datasets where each one belongs to each technique. For random and clustering under-sampling techniques, we have used *imblean.under_sampling* python package. For high-length and low-length under-sampling, we have added a “length” attribute. Then, we have sorted the tweets using the length attribute. Then, we have extracted the required size from each category and sub-category.

Table 58 shows the results of under sampling for data that do not have emojis while Table 59 represents the result of under-sampling for data that have emojis.

Table 58: Under-sampling output distribution for tweets without emojis

groups	mix				Mixed categories- in one polarity		pure categories-related							
total size	446 tweets				686 tweets		1680 tweets							
sub-groups	equal score	equal score	dominated by fanatic	dominated by non-fanatic	fanatic	non-fanatic	adaptable	tolerance	respecting	Knowledge	passion	hatred	aggression	agitation
# tweets	61	61	162	162	343	343	210	210	210	210	210	210	210	210
Validated as	-1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1

Table 59. Under-sampling output distribution for tweets with emojis

groups	mix				Mixed categories- in one polarity		pure categories-related							
total size	114 tweets				250 tweets		512 tweets							
sub-groups	equal score	equal score	dominated by fanatic	dominated by non-fanatic	fanatic	non-fanatic	adaptable	tolerance	respecting	Knowledge	passion	hatred	aggression	agitation
# tweets	9	9	48	48	125	125	64	64	64	64	64	64	64	64
Validated as	-1	1	-1	1	-1	1	1	1	1	1	-1	-1	-1	-1

In this section, we have presented the extraction of eight test datasets: four of them have text with emojis and four of them have text without emojis. The four that have emojis will be used to test the emojis-lexicons and emojis-classification-models.

As an outcome of this process, the test datasets are balanced and cover most of the cases required to measure the classification models.

CHAPTER 8

FEATURE ENGINEERING

This chapter presents the conducted feature engineering work to convert tweets' text into useful features. The data have been formed into suitable representations for machine learning models. Feature engineering is a process of turning the raw data into efficient and suitable input for machine learning models. This process requires understanding the domain-knowledge and the phenomenon being observed. Zheng and Casari have defined feature engineering as “Feature engineering is the act of extracting features from raw data and transforming them into formats that are suitable for the machine learning model.” [85]. Feature engineering consists of tasks that include feature identification, feature extraction, feature transformation, feature selection, and feature evaluation [87]. In this work, feature engineering is categorized into three tasks: feature-identification, feature-construction, and feature-reduction. These tasks are illustrated in Figure 39.

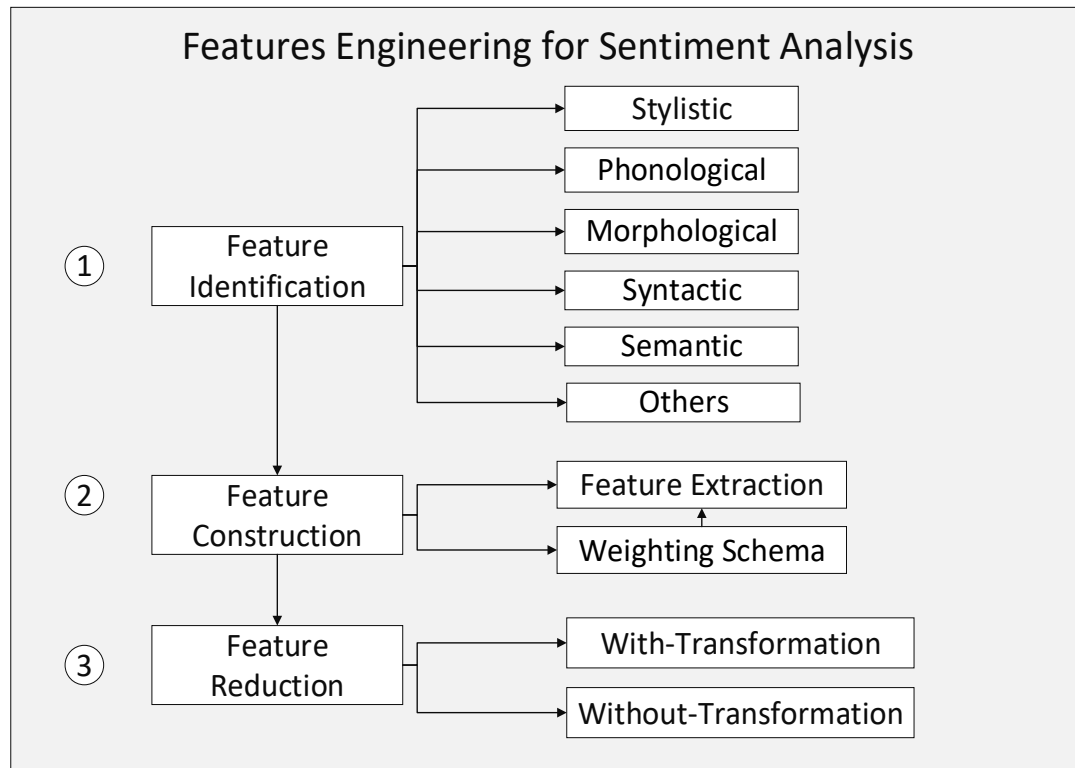


Figure 39. Feature engineering stages for sentiment analysis

Feature-identification: in this stage, domain-knowledge and data-analytic are required. We started this stage by understanding the problem and analyzing the data. Then, we studied how feature engineering is conducted on other close-domain problems. Then, we conducted brainstorming to decide what are the suitable features related to problem-domain. To identify some suitable feature-sets for Arabic sentiment analysis, we would like first to refer to what the researchers in computational linguistics (CL) have done to arrange the stages of linguistic analysis. Linguistic analysis has four levels: phonological level, morphological level, syntactic level and semantic level. Each level has some aspects of the language. Hence, some features that have a positive impact on sentiment analysis can be extracted from each level. The intended features can be categorized into two categories: content-related features and container-related features. The content-related feature sets are: phonological feature set, morphological feature set, syntactic feature set and semantic

feature set. On the other hand, the container-related features are features that are related to the style of contents. We will present more details on each one of these sets in the next section.

Feature-construction: in this stage, the identified-features are extracted with suitable weighting schema. For example, bag-of-words feature could have a binary weighting schema where each word in the document can be represented in the vector space as 0 or 1. Correspondingly, it could also take a word-count, TFIDF, or a weight that is extracted from senti-lexicon. In sentiment analysis, feature construction requires resources such as senti-lexicons, dictionaries, ontologies, morphological analyzer, stemmer, normalizer, tokenizer, special-purpose programs and/or other resources.

Feature-reduction: the purpose of this stage is to improve the performance by removing redundant or/and irrelevant features. This task has usually a positive impact on performance. In addition, it could help in removing features that have a negative impact on the accuracy. There are two types of reduction: reduction with transformation and reduction without transformation. For the first one, the features are reduced by transforming the features into other type of features such as reduction using *Principal Components Analysis* (PCA). For the second type (reduction without transformation), the features are reduced by selecting a suitable subset such as selection using weka's CfsSubsetEval which evaluates both irrelevant features and redundant features. Evaluation for irrelevancy could help how relevant are given features to the positive and negative class (assuming we have only two classes). Evaluation for redundancy could help eliminate the redundant features by examining the correlations among features.

8.1 Feature Identification and Extraction

In this section, we present the five feature sets that we have used which are: phonological features, morphological features, syntactic features, semantic features and stylistic features.

8.1.1 Phonological Features

This type of features could be considered as low-level features because it cares about determining the characteristics of the pronunciation of the letters and their groups. Tajweed science helps us to extract some features that might show useful meaning in sentiment analysis. What are the features that can be extracted from this level? We can extract some of low-level features such as: Vowels, Diacritics, lengthening, Articulation-Points related letters (e.g throat Letters: ء ه ع غ ح خ), Al-saffir letters (س ص ز), elevation/bold/strong letters (خ ص ض غ ط ق ظ), lowering/light/soft letters (ظ) and other features. For example, we can examine how elevation-letters are related to fanatic and non-fanatic reviews. We need to check if there is a hidden semantic that may play some roles on the domain-problem. Do Arabic words have special sentiment-related meaning in this level? These features have not previously examined for sentiment analysis. Going from low-level features (phonemes) to high-level (words and sentence) features and extracting the hidden semantics, can be considered as white-box deep learning.

In this level, 24 groups of letters have been examined as primary testing to check their impact on fanatic and non-fanatic words. We expect that the selected-features may carry some semantic related to the domain-problem. For example, one assumption is that Arabic words with more elevation letters may carry fanatic while lowering letters may not. To

examine such assumption, we have conducted some statistics on our manual fanatic-lexicon where we have two word-groups: fanatic-words and not-fanatic words. Then, we have calculated the ratio of elevation and lowering letters in both groups. Table 60 and Figure 40 show the summary of the calculation of all proposed features. The expectation is that these features might have small contribution when building the model. Examining such features helps to understand some aspects of Arabic language. In other words, we examine if there is a relation between Articulation-Points of Arabic letters and the fanatic-words.

Table 60. Proposed phonological features

Phonological Features	Letter Ratio in lexicon words		letters
	Not fanatic	fanatic	
Alsaffir letters ratio	0.021	0.046	س ص ز
Elevation letters ratio	0.056	0.099	خ ص ض غ ط ق ظ
Lowering letters ratio	0.845	0.832	ا ب ت ث ج ح د ذ ر ز س ش ع ف ك ل م ن ه و ي
Qlqlah_letters ratio	0.124	0.135	ق ط ب ج د
Lips letters ratio	0.175	0.201	ب م و ف
Lips1 letters ratio	0.144	0.178	ب م و
Lips2 letters ratio	0.030	0.023	ف
Nose letters ratio	0.116	0.145	ن م
Oral cavities ratio	0.201	0.204	ا و ي
Throat letters ratio	0.108	0.155	ء ه ع ح غ خ

Phonological Features	Letter Ratio in lexicon words		letters
	Not fanatic	fanatic	
Throat1 letters ratio	0.052	0.059	ء ه
Throat2 letters ratio	0.048	0.049	ح ع
Throat3 letters ratio	0.009	0.046	غ خ
Tongue letters ratio	0.454	0.405	ق ك ج ش ي ت ط د ث ظ ذ ن ر ز ص س ض ل
Tongue1 letters ratio	0.028	0.020	ق
Tongue2 letters ratio	0.040	0.020	ك
Tongue3 letters ratio	0.058	0.030	ج ش ي
Tongue4 letters ratio	0.065	0.089	ت ط د
Tongue5 letters ratio	0.003	0.013	ث ظ ذ
Tongue6 letters ratio	0.038	0.053	ن
Tongue7 letters ratio	0.070	0.079	ر
Tongue8 letters ratio	0.021	0.046	ز ص س
Tongue9 letters ratio	0.008	0.000	ض
Tongue10 letters ratio	0.123	0.056	ل

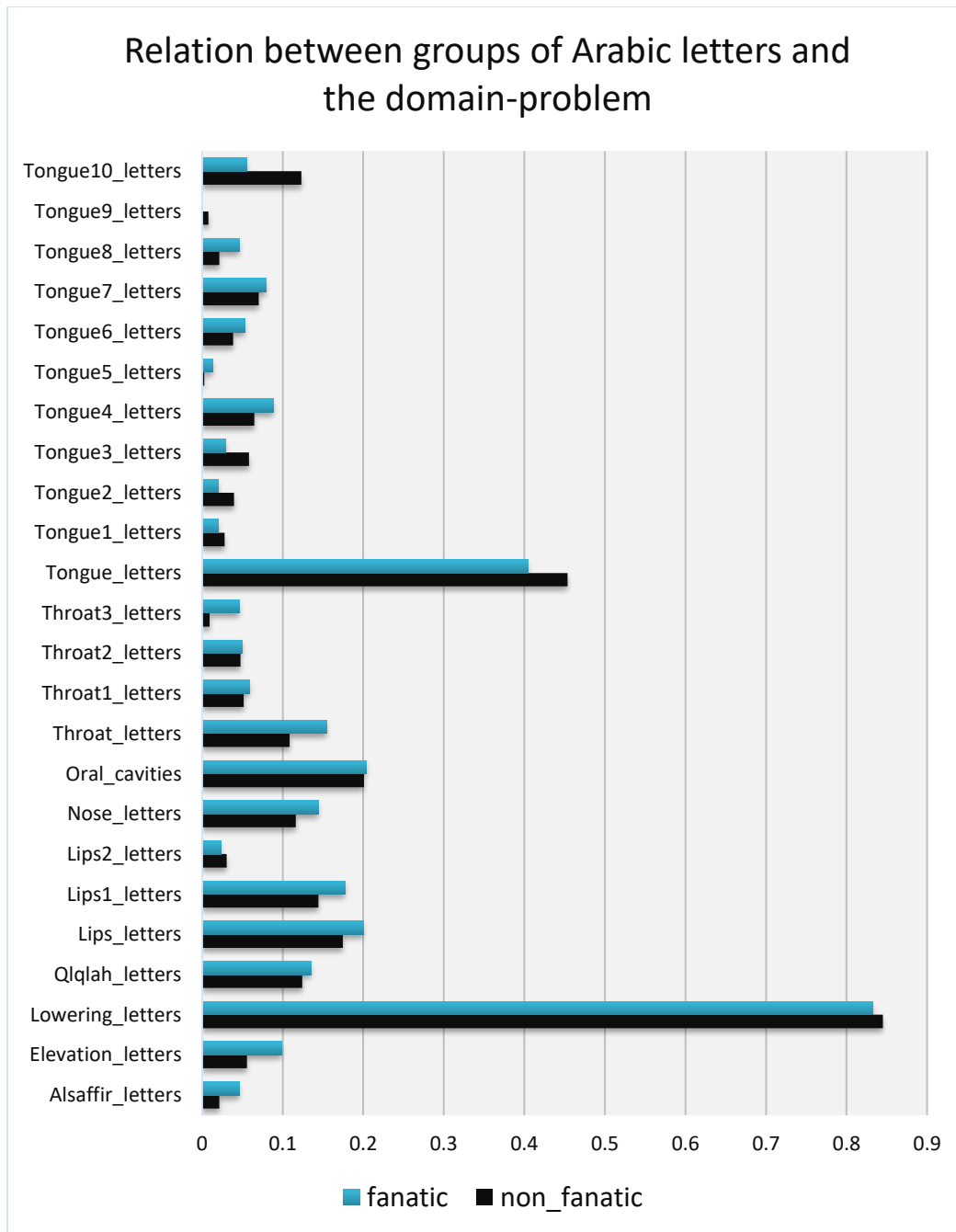


Figure 40. The impact of phonological features on domain-problem

We can observe that the groups of letters that come out from throat, nose and lips articulation-points, are more related to fanaticism words. On the other hand, the letters that come out from Tongue articulation-point, are more associated to non-fanaticism words. The letters that come out from oral cavities articulation-point, have almost the same ratio

in both fanaticism and non-fanaticism words. For lowering and elevation letters, we can see that the lowering letters are more related to non-fanaticism words with slightly small difference while the elevation letters are used in fanaticism's words double compared with the use of non-fanaticism. Although both Alsafir and Qlqlah letters are more related to fanaticism words, Alsafir letters shows double difference while the Qlqlah letters show slightly small difference. Generally, we can deduce that the 24 features extracted from this level can add small contribution when building the proposed classification-model. To extract the identified phonological features for document-classification, a java program was developed to compute the ratio of each feature where the occurrence of each feature's letters in each document is divided by the total number of letters in the document.

8.1.2 Morphological features

While the previous level deals with phonology of letters, this level goes up and deals with different segments of a word including its affixes. In this stage, a word is analyzed into its basic elements (prefix, stem and suffix). The possible extracted features in this level could be bag-of-segments feature representation, bag-of-stems feature representation, bag-of-lemma feature representation, bag-of-root feature representation, suffixes count (e.g countSuffixes), prefixes count (e.g countPrefixes), and other related features. The weighting schema could be binary, count, ratio or TFIDF score.

Many researches in Arabic sentiment analysis use bag-of-stems generated by a light stemmer such as Madamira. This form has shown good results. However, we believe that some affixes might have a small effect on Fanatic-classification. For example (لعب) and (لعبكم), the suffixes with person pronoun might show an indicator of personalities the

discussion (شخصنة النقاش). For this, we might need to also examine bag-of-normalizedWords feature representation.

To extract the identified morphological features for document-classification, we have utilized Farasa stemmer to do stemming (See Chapter 6). After that, we have applied vectorization to convert the preprocessed string to word vectors where each word is represented by long sparse vector as shown in Table 35. To apply vectorization, two Weka's filters were applied (*fromStringToWordVector* and *FixedDictionaryStringToWordVector*). Then, different classification algorithms were examined. More information about classification is presented in Chapter 9. Different weighting schema can be applied such as binary, count and TFIDF score.

8.1.3 Syntactic Features

While the previous level analyzes the structure of a word. In this level, we can examine the function of the word in the sentence. In this stage, we study the relation among words through part-of-speech (POS) tagging. This level might include extracting some features such as POS Tags, phrase patterns, sentence boundaries, punctuation, negation words, intensifier words, diminisher words, modal words, etc. The weighting schema could be binary, count, ratio or TFIDF score. Therefore, 24 syntactic features were identified as shown in Table 61. These features contain the negation, conjunction, and other features. To extract these features, we have built 19 dictionaries that contain the words related to each feature. For example, we have built a dictionary that contains negation words and negation patterns. All dictionaries were collected manually from different websites. Then, we have expanded the dictionaries' words using google translation where the synonyms

were manually extracted. Then, we have built a java program to extract the count and the ratio of these features for each tweet. These features could help in answering the following questions: Is there a relation between different types of transient words and problem domain? What is the effect of personal-pronouns and demonstrative-pronouns on fanatic? What is the impact of intensifier and modal words? What is the impact of negation words?

Table 61. Identified syntactic features

Features Metrics	Weighting-Schema used	Metrics Description	Features Example
Personal pronoun ratio	Ratio	Computing the percentage of the personal pronouns in each tweet.	انت، هي، هو
Conjunction terms	Count	Counting the occurrences of conjunctions in each tweet.	و، او، ف
Conjunction terms ratio	Ratio	Computing the percentage of the conjunctions in each tweet	و، او، ف
Negation pattern	Count	Counting the occurrences of negation patterns in each tweet	ماشلعيش -> ماش***ش
Negation pattern ratio	Ratio	Computing the percentage of the negation patterns in each tweet.	ماشلعيش -> ماش***ش
Negation words	Count	Counting the occurrences of negation words in each tweet	ليس، لا، لن، مو، مش
Negation words	Ratio	Computing the percentage of negation words in each tweet	ليس، لا، لن، مو، مش
Negation words and patterns	Ratio	Computing the percentage of the negation in each tweet	ماشلعيش -> ماش***ش ليس، لا، لن، مو، مش

Features Metrics	Weighting-Schema used	Metrics Description	Features Example
Addition phrases	Count	Counting the occurrences of addition phrases in each tweet	ايضاً، مراراً، مجدداً، وكذلك
Consequence phrases	Count	Counting the occurrences of consequence phrases in each tweet	وفقاً لذلك، كنتيجة، بناء عليه
Contrast phrases	Count	Counting the occurrences of contrast phrases in each tweet	لكن، من ناحية أخرى، في الجانب الآخر
Demonstrative pronoun terms	Count	Counting the occurrences of demonstrative pronouns in each tweet	هذا، هذه، ده، ديه
Direction phrases	Count	Counting the occurrences of direction phrases in each tweet	هنا، هنالك، وراء، فوق
Diversion phrases	Count	Counting the occurrences of diversion phrases in each tweet	على فكرة، بالمناسبة
Emphasis phrases	Count	Counting the occurrences of emphasis phrases in each tweet	بالدرجة الأولى، خصوصاً، بشكل رئيسي
Exception phrases	Count	Counting the occurrences of exception phrases in each tweet	إلا إذا، إلا، في حال، ماعد، باستثناء
Exemplifying phrases	Count	Counting the occurrences of exemplifying phrases in each tweet	أعني، مثلاً، على سبيل المثال
Generalizing phrases	Count	Counting the occurrences of generalizing phrases in each tweet	كقاعدة، كقانون، عادةً، بصفه عامة

Features Metrics	Weighting-Schema used	Metrics Description	Features Example
Intensifier phrases	Count	Counting the occurrences of intensifier phrases in each tweet	جداً، تماماً، للغاية
Modal phrases	Count	Counting the occurrences of modal phrases in each tweet	ضروري، ممكن، لازم، يجب
Restatement phrases	Count	Counting the occurrences of restatement phrases in each tweet	بعبارة أخرى، بالمختصر، وهذا يعني
Sequence phrases	Count	Counting the occurrences of Sequence phrases in each tweet	أولاً، بدايئاً، بعد ذلك، التالي
Similarity phrases	Count	Counting the occurrences of similarity phrases in each tweet	بشكل مماثل، متطابق، يتطابق
Summarizing phrases	Count	Counting the occurrences of summarizing phrases in each tweet	بعد كل شيء، كنتيجة، في الختام

8.1.4 Semantic Features

While the previous level analyzes the structure of the sentence and the relation between words, semantic-features level analyzes what the meaning of a word in a context. This level has a direct relation to the domain-problem. Here, we can decide how the words are related to fanatic and non-fanatic groups. In this stage, we extract information related to the meaning of the content from senti-lexicons. We can extract related features such as count of fanatic emojis, score of non-fanatic emojis, lexicon score of fanatic or non-fanatic emojis, count of non-fanatic words, score of fanatic words, lexicon score of fanatic or non-

fanatic words, ratio of non-fanatic words, list of lexicon terms (bag-of-lexiconTerms), count of fanatic sentences and other features. Therefore, we have used our fanatic-lexicons to construct 32 features. Details about constructed lexicons was discussed in Chapter 7. We have extracted 8 indicators-related features where the degree of non-fanatic and the degree of fanatic was extracted from lexicons. Table 62 shows some of the extracted features using automatically generated lexicons.

Table 62. Identified semantic features

Extracted Features	weighting-schema
Adaptable phrases	Count
Aggression phrases	Count
Agitation phrases	Count
broad-minded phrases	Count
Hatred phrases	Count
Tolerance phrases	Count
Passionate phrases	Count
Respecting phrases	Count
Non-fanatic phrases	Ratio
Fanatic phrases	Ratio
Confidence Predicted value by lexicon-based classification	Ratio
Predicted-label by lexicon-based classification	Binary
Lexicon words used as Bag-of-words	Binary, count, TFIDF score

8.1.5 Stylistic Features

Stylistic features are external features describe the container and the style of the content. These features could have some indicators about the emotion and the feeling. The writing style including used colors, font size might give an impression about the feeling of the writer. Here, we can extract some features such as word count, character count, lines count, vocabulary richness measures, special character persistence, digit persistence, red color persistence and other features. In this work, we have extracted six features related to the document size. We need to test if there is a relation between the size of the document and the fanaticism. We have built a java program to extract these features. Table 63 shows the details of these stylistic features.

Table 63. Identified stylistic features

Features	Weighting-schema used	Description/Examples
Characters count	Count	Number of characters in each tweet
Words count	Count	Number of words in each tweet
Unique words count	Count	Number of unique words in each tweet
Emojis count	Count	Number of emojis in each tweet
Unique emojis count	Count	Number of unique emojis in each tweet
Average word length	Ratio	Average length of the words in each tweet

8.2 Feature Reduction

In this work, we have applied features-reduction through four ways: morphologically reduction, removing high frequent words (stop-words), removing low frequent words (typos and rare words) and applying feature selection techniques to select relevant features.

Morphologically reduction helps in reducing several forms of the same word into less possible forms. While each word in the text is represented as on bag-of-words feature representation, it is possible that the same word with different forms can have number of features as number of forms. For Example, without morphological reduction, each word of the following list (والفريقين, بالفريقين, بالفريقين, بالفريقين) can each of these words represented with a separate feature. While all of these words are about one feature (فريق). Morphological analysis usually helps to form all such cases into a smaller number of forms with less features. The morphological analysis was discussed in chapter 6. In this work, we have applied lemmatization and light stemming using Farasa stemmer discussed in Section 6.3.

We have also conducted feature reduction by removing high frequent stop-words. Stop-words are words that are always used in both positive and negative documents. They do not usually have a semantic meaning to the domain-problem. For example, words like (في، من) are frequently used in both positive and negative documents where they are not relevant to the classification problem. Therefore, we can remove those stop-words to reduce the number of features. In our work, some stop-words were extracted from our dataset and some were collected from two stemmers: *khoja-stemmer* and *lucene-arabic-analyzer*. The collected stop-words were filtered manually. We have analyzed our data and selected some

of the high frequent words that show the same degree of frequency in all contexts of our data. We ended up with 151 stop-words.

We also conducted feature reduction by removing low frequent words. Those words could be typos or rare words. To remove low frequent words, we made minimum threshold where any word appears in documents less than the proposed threshold will be ignored. We have examined five thresholds:10, 20, 30, 40, 60 and 100 as minimum frequency. We have selected high-scale thresholds because the size of the training dataset is quietly large where our training dataset have 276176 tweets. When the threshold is high, it allows the probability models such as Naïve Bayes (NB) to produce more intuitive statistical values.

For feature selection techniques, there are some techniques that can be used to remove irrelevant and/ or redundant features. One of these methods is information gain method which removes irrelevant features (e.g InfoGainAttributeEval in WEKA). Other methods have the ability to remove both irrelevant and redundant features (e.g CfsSubsetEval in WEKA). In bag-of-words feature representation, we have found that removing redundant features may not be a good idea as it removes important words. For example, the two words (ضفادع) and (طحالب) are used in one of the fanatic-context (contempt), where each word is represented as a feature. These two features might appear together. Therefore, removing redundant words may result in removing one of them where both are important in our work. For this reason, we have chosen a feature selection technique that only removes irrelevant features. In this case, Chi-Square feature-selection technique (e.g ChiSquaredAttributeEval in WEKA) is applied to select relevant features. Chandra and Gupta [132] studied the performance of seven statistical feature selection techniques for classification on six well-

known gene expression datasets. Their results showed that the Chi-squared performs a little better than information-gain.

CHAPTER 9

CLASSIFICATION AND EVALUATIONS

In this chapter, we present our experiments and findings using three classification approaches. These approaches are rule-based approach, learning-based approach and hybrid approach. In rule-based classification approach, we used the developed lexicons to conduct classification, and we evaluated the performance of the developed resources. In learning-based approach, we used machine learning algorithms to conduct classification. In the hybrid classification approach, we combined both rule-based and learning-based to conduct classification.

9.1 Experimental Setup

In this section, we describe the general setting that we have applied in our experiments. We also provided details on the used datasets for training and testing, the utilized tools, the classifiers, the used parameters, and the evaluation metrics.

9.1.1 Training dataset

We have preprocessed 276176 tweets and run Farasa stemmer on them, as described in chapter 6. The produced corpus contains 86794 non-fanatic tweets and 189382 fanatic tweets. The methods of collecting the items of the corpus and the annotation process are described in Chapter 7. While the original prepared training dataset was imbalanced, we have made it balanced through random sampling (Using Weka, we have utilized the supervised filter called *speroadSubsample* with value (1) for parameter

distributionSpread). As a result, our updated training dataset has 86794 non-fanatic tweets and 86794 fanatic tweets. For emojis-classification, we have extracted tweets that have emojis from the original corpus (276176 tweets). We have 21929 non-fanatic tweets and 21929 fanatic tweets. The extracted tweets are used as our balanced training set for emojis-classification.

9.1.2 Test dataset

To test and validate each learning-based classification model, we have used either cross-validation (5-fold/10-fold), or dataset splitting: 90% for training and 10% for testing. In addition, we have used balanced test datasets that have been mainly constructed for testing. These test datasets were separately constructed and annotated. The approach of building the test datasets were discussed in Section 7.4. For text, the test dataset contains 3688 tweets distributed in 8 contexts. For emojis, we have applied the same approach where the classification models have been evaluated twice. In one of them, we have prepared a test dataset where each tweet has at least one emoji. While the test datasets were extracted by applying four under-sampling techniques as discussed in Section 7.4, we used the one that has been resulted by applying random under-sampling.

9.1.3 Tools

We have utilized *Weka* to conduct our classification experiments. *Weka* stands for Waikato Environment for Knowledge Analysis [110]. It is a machine learning platform that provides graphical user interface to conduct data analysis and predictive modelling. It contains a collection of tools and algorithms for classification, clustering, feature selection and data preprocessing. It is available freely under GNU General Public License. We have selected

Weka because it has the classification tools needed for our research work. We have installed it on a computer with 8 GB RAM and CORE i7 processor.

9.1.4 Classifiers

For classification, we have applied Naïve Bayes (NB), Logistic Regression (LR) and Support Vector Machine (SVM). NB classifiers are probabilistic classifiers that are based on Bayes' theorem. NB based classifiers are popular in sentiment analysis and have been used in several studies such as the work of Mountassir et al. [133] and the work of Al-Moslmi et al. [134]. For implementation issues, we have used the Naïve Bayes class in *Weka* (*weka.classifiers.bayes.NaiveBayes*). While we have training corpus with around 300K, the probabilistic classifiers are good choice because large corpora usually provide to the classifiers more information to conduct statistics and deduce good probabilities. NB can predict the class of a given document by calculating the probability of each term in the document under the assumption that each term is independent from other terms. In general, it is assumed that the used features are independent. In bag-of words feature representation, the order of the words is not important, and each word is represented as an independent feature. For this, we have used bigrams and trigrams to keep some word dependency in a way that the dependencies are encapsulated and represented as independent features. NB works as follows: given a collection m of N documents $D = \{d_1, d_2, \dots, d_n\}$ where each document d_i is represented as a sequence T of m terms $T = \{t_1, t_2, \dots, t_m\}$, where t_i is a term in the sequence. The probability of a document $d_i \in D$ occurring in class $c_k \in C$ where $C = \{poistive, negative\}$, is calculated as follows:

$$P(c_k | d_i) = P(c_k) \prod_{j=1}^m p(t_j | c_k) \quad (14)$$

Where $p(t_j | c_k)$ is the probability of term t_j occurring in documents of class c_k . $P(c_k)$ is the prior probability of documents with class c_k which can be estimated from the training dataset.

For Logistic regression and SVM, we have used a *Weka* library for large linear classification [135]. The name of the used library is *LibLINEAR* (*weka.classifiers.functions.LibLINEAR* in Weka). It is an open sources library that provides efficient implementation for LR and SVM on large sparse datasets such as text classifications. *LibLINEAR* is efficient for large-scale training datasets [135]. It provides L1 & L2-regularization as techniques to deal with overfitting. We have used L2-regularization because L1 leads to sparse output. While our features are words, we do not need to eliminate them. L1-regularization could work as a feature selection tool. For evaluation metrics, we have used precision, recall, accuracy and F-measure. These metrics are common metrics used in the literature to evaluate 2-class classifications.

9.1.5 Evaluation Metrics

Model evaluation helps to find the best model of our data and shows how well the model will work in future. To avoid overfitting, we have evaluated each classification-model twice using two test sets. The first evaluation was by splitting a corpus with 300K tweets into (90% training and 10% testing). The second evaluation was using a second corpus that was prepared separately. We use the model evaluation metrics to compare the predicted classes against actual classes. In binary classifications, confusion matrix is used to show

the correct predictions and incorrect predictions achieved by the classification model. Table 64 shows a confusion matrix for binary-classification that contains four cells: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [136]. In our work, the confusion matrix can be defined as follows:

- True Negative (TN) is the number of documents that are non-fanatic and predicted to be non-fanatic.
- True Positive (TP) is the number of documents that are fanatic and predicted to be fanatic.
- False Negative (FN) is the number of documents that are fanatic but predicted to be non-fanatic.
- False Positive (FP) is the number of documents that are non-fanatic but predicted to be fanatic.

Table 64. Confusion Matrix for domain-problem

Confusion Matrix		Predicted class	
		Fanatic	Non-fanatic
Actual class	Fanatic	TP	FN
	Non-fanatic	FP	TN

We used confusion matrix to perform the following five evaluation metrics. The five-evaluation metrics used to measure the performance of the proposed classification models. The used evaluation metrics are specificity, recall, precision, accuracy and F1-score.

Specificity is also called (True Negative Rate). It is the ratio of correctly predicted non-fanatic-documents to all non-fanatic-documents. Specificity can be calculated as follows:

$$specificity = \frac{TN}{TN + FP} \quad (15)$$

Specificity

$$= \frac{\text{nonfanatic correctly identified}}{\text{nonfanatic correctly identified} + \text{nonfanatic incorrectly labeled as fanatic}}$$

Recall is also called (True Positive Rate). It is the ratio of correctly predicted fanatic-documents to all fanatic-documents. Recall can be calculated as follows:

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$Recall = \frac{\text{fanatic correctly identified}}{\text{fanatic correctly identified} + \text{fanatic incorrectly labeled as nonfanatic}}$$

Precision is the ratio of correctly predicted fanatic documents to the total predicted fanatic documents. Precision can be calculated as follows:

$$precision = \frac{TP}{TP + FP} \quad (17)$$

$$precision = \frac{\text{fanatic correctly identified}}{\text{fanatic correctly identified} + \text{nonfanatic incorrectly labeled as fanatic}}$$

Accuracy is the ratio of correctly predicted documents to the total documents. It answers the following question: What is the percentage of the predicted documents that were correctly identified? Accuracy can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

F1-Score is also called F1-measure. It is a single metric that combines both recall and precision using harmonic mean. It is the weighted average of precision and recall. It helps to decide which classification model is better. When we have two models or more and we are confused to decide which one is better where one has higher recall and the other has higher precision. In such case, F1-Score provides a single value that helps to decide. F1-Score can be calculated as follows:

$$F_1\text{score} = 2 * \frac{\textit{precision} * \textit{recall}}{\textit{precision} + \textit{recall}} \quad (19)$$

In the binary-classifications, the intuition of specificity, recall and precision is that it solves some limitations of accuracy metric. They provide more information to decide which model is better. For example, accuracy just shows the performance of all predictions of both classes, but it does not tell in which class the model performs better. One of the weaknesses of accuracy metric is a bias to majority class [136]. Accuracy metric does not answer the following question: Is the model doing better on a fanatic class or on a non-fanatic class? Therefore, the recall metric shows the performance of a model on the fanatic class and the specificity metric shows the performance of a model on the non-fanatic class. While the accuracy metric cannot tell the ratio of the model performance on the predictions on fanatic documents to all predictions, the precision can handle this limitation.

9.2 Rule-Based Classification Approach

Rule-based classification is sometimes called lexicon-based approach as a lexicon is used to conduct classification. Here, we will evaluate two types of lexicons: a manually extracted lexicon, and automatically generated lexicons.

9.2.1 Classifications Using a Manual Lexicon

In this scenario, the manually constructed fanatic-lexicon is evaluated. To conduct evaluation, first we have used this lexicon to automatically annotate a corpus with around 11K tweets. The annotated corpus is constructed under two requirements: high-coverage and low-redundancy. The purpose of these two requirements is to cover most of possible cases, contexts and context-related phrases. We have used the method described in Section 7.4 to construct the corpus. Secondly, we have developed a web-application tool to simplify analysis, annotation and validating the annotated corpus. A Demo of this tool is presented in the video at (<https://youtu.be/9CQaVTRA6D8>) [131]. We have performed manual validation and annotation by ourselves. The results were presented in Table 54 and Table 55 of Section 7.4. The results show 93.40% accuracy. We have found that 10600 tweets were correctly annotated by the lexicon and 749 tweets were not.

9.2.2 Classifications Using Automatically Generated Lexicons

In this scenario, we have compared the performance of the two techniques: PPMI and proposed-TFICF. These two techniques were used to build seven fanatic-lexicons (See in Section 7.2). In this scenario, we have evaluated 7 types of lexicons: unigrams of normalized text, unigrams of stemmed text, bigrams of stemmed text, trigrams of stemmed text, unigrams of emojis, bigrams of emojis, and trigrams of emojis. We were looking for

answers of the following questions: which technique performs better: (PPMI or the proposed-TFICF)? Which lexicon-type performs better: unigrams vs bigrams, text vs emojis and normalized text vs stemming text?

9.2.2.1 Classification Method

Each term in the lexicon has eight scores: four non-fanatic indicators and four fanatic indicators (See Section 4.1). Therefore, the given tweet is tokenized into suitable terms (unigrams, bigrams or trigrams). Each term is passed as a search string in order to find its score in the lexicon. The non-fanatic and the fanatic scores of all tokens are summed up and the predicted label of the tweet is determined. If the overall non-fanatic score is higher than the overall fanatic score, then the predicted label is assumed to be non-fanatic. If the overall fanatic score is higher than the overall non-fanatic score, then the predicted label is assumed to be fanatic. If the terms (tokens) of the tweet are not covered by the lexicon, or the fanatic and the non-fanatic scores are equal, then the predicted label is assumed to be zero. While test sets have only two labels, the zero-prediction is considered as a false positive or a false negative. Figure 41 shows the proposed rule-based classification approach.

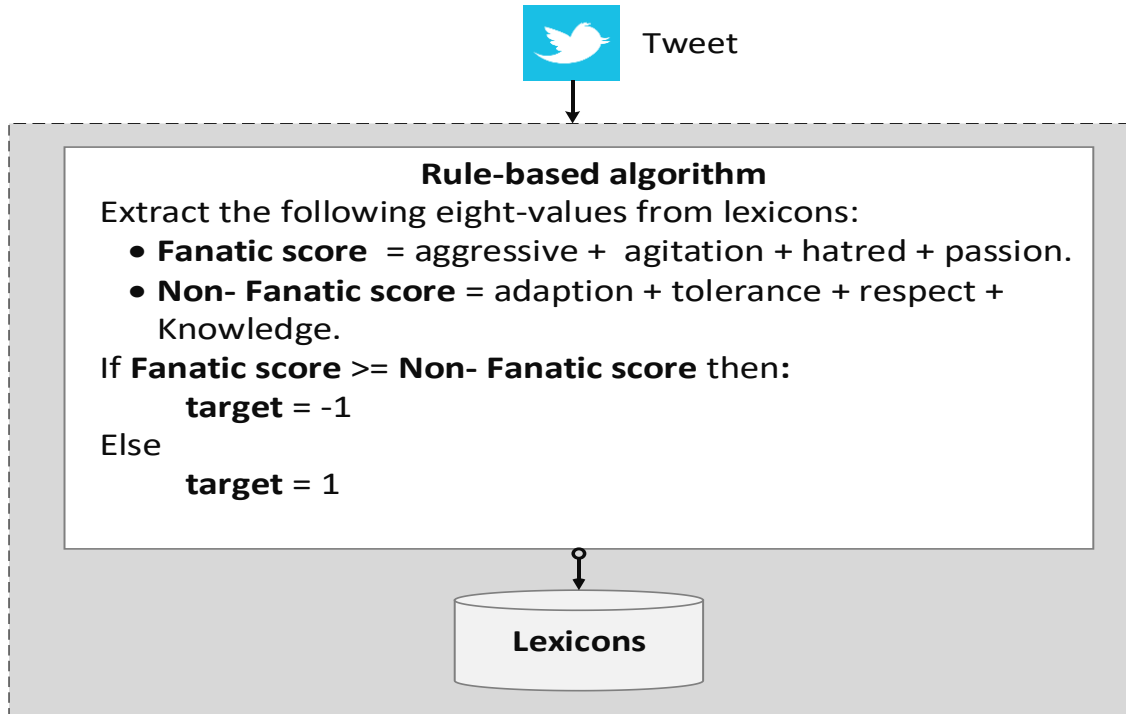


Figure 41. Rule-based classification approach

9.2.2.2 Tools

We have developed a tokenizer to generate and count ngrams terms. We have also implemented a tool to compute rule-based classification metrics: accuracy, precision, recall and f1-score. We have utilized MongoDB database to store and quarry lexicons. The lexicons were used for prediction and the result of evaluation was stored. As a noticeable point, using database indexing for lexicon terms has improved the performance of searching.

9.2.2.3 Experiments Setup and Outcomes

we have used four test datasets that are not parts of the corpus that we have used to build the lexicons. The test datasets were discussed in Section 7.4. For emojis lexicons, four balanced test datasets were used for testing (See Table 59). For text lexicons, four balanced datasets were used for testing (see Section 7.4). Each test dataset contains 3688 tweets.

Table 65 shows the impact of under-sampling techniques on performance. Four under-sampling techniques were examined which are: cluster centroids sampling, Random sampling, choosing the highest length documents, and choosing the lowest length documents. Figure 42 reflects what are presented in Table 65. It illustrates the results of applying the four under-sampling techniques and shows the impact of two lexicons. The two lexicons were generated by different methods where one generated by PPMI and the other generated by TFICF as explained in Section 7.2. It can be seen that the tweets with low-length performed better than the tweets with high-length. This might be because a low length tweet is limited to one opinion where the fanatic phrases can be detected and highlighted. The high length tweet might have non-fanatic and fanatic sentiments which make it difficult for the classifier to decide. We can also observe that the lexicons generated by the proposed-TFICF performed better than the lexicons generated by PPMI with all under sampling datasets.

Table 65. The impact of different techniques for under-sampling on performance

Lexicon used	Test datasets under-sampled by	Accuracy	Precision	Recall	F1-score
Unigrams of stems generated by PPMI Size 24438	Cluster centroids	0.768	0.695	0.958	0.805
	Random	0.797	0.726	0.954	0.825
	High length	0.735	0.663	0.957	0.783
	Low length	0.865	0.812	0.950	0.875
Unigrams of stems generated by Proposed-TFICF Size 24438	Cluster centroids	0.874	0.832	0.937	0.881
	Random	0.890	0.857	0.935	0.894
	High length	0.861	0.814	0.937	0.871
	Low length	0.915	0.901	0.931	0.916



Figure 42. The performance of under-sampling techniques

Table 66 and Table 67 show the impact of different data type (Text and emojis), different preprocessing level (normalizing and stemming) and different token types (unigrams, bigrams and trigrams). To simplify the summary of the results for these two tables, we computed the average result of the four test datasets under-sampled by the four under-sampling techniques. While Table 66 presents the results of the lexicons that generated by PPMI method, Table 67 shows the results of applying lexicons that generated by TFICF method. Figure 43 and Figure 44 illustrate what are displayed in Table 66 and Table 67 respectively. From both figures, we can observe that unigrams and bigrams lexicons of text datatype have achieved higher accuracy than trigrams. This is because that many tweets in the test datasets were not covered by trigrams lexicon terms and were considered as false positive or false negative. We can also see that unigrams of emojis has performed better than bigrams and trigrams for the same reason discussed in previous sentence. We can notice that the normalized text has performed a little bit better than stemmed text. This might be because some prefixes and suffixes have impact on fanatic, or some positive and

negative words have overlapped after stemming. This might need more study and analysis which will be considered as a future work. Moreover, we can observe that the lexicons generated by the proposed-TFICF (Table 67) performed better than the lexicons generated by PPMI (Table 66) in all experiments.

Table 66. The performance of seven fanatic-lexicons generated by PPMI method

Token-type	Data type	Precision	Recall	Accuracy	F1-score
Normalized unigrams without stop-words	Text	0.760	0.937	0.817	0.838
Normalized unigrams: 80374	Text	0.743	0.948	0.806	0.832
Unigrams of stems: size 24438	Text	0.729	0.950	0.794	0.823
Bigrams of stems: 155004	Text	0.752	0.905	0.803	0.821
Trigrams of stems: size 80374	Text	0.580	0.658	0.588	0.616
Unigrams with size 692	Emojis	0.651	0.777	0.678	0.708
Bigrams with size 1714	Emojis	0.412	0.445	0.405	0.428
Trigrams with size 1384	Emojis	0.170	0.162	0.184	0.166

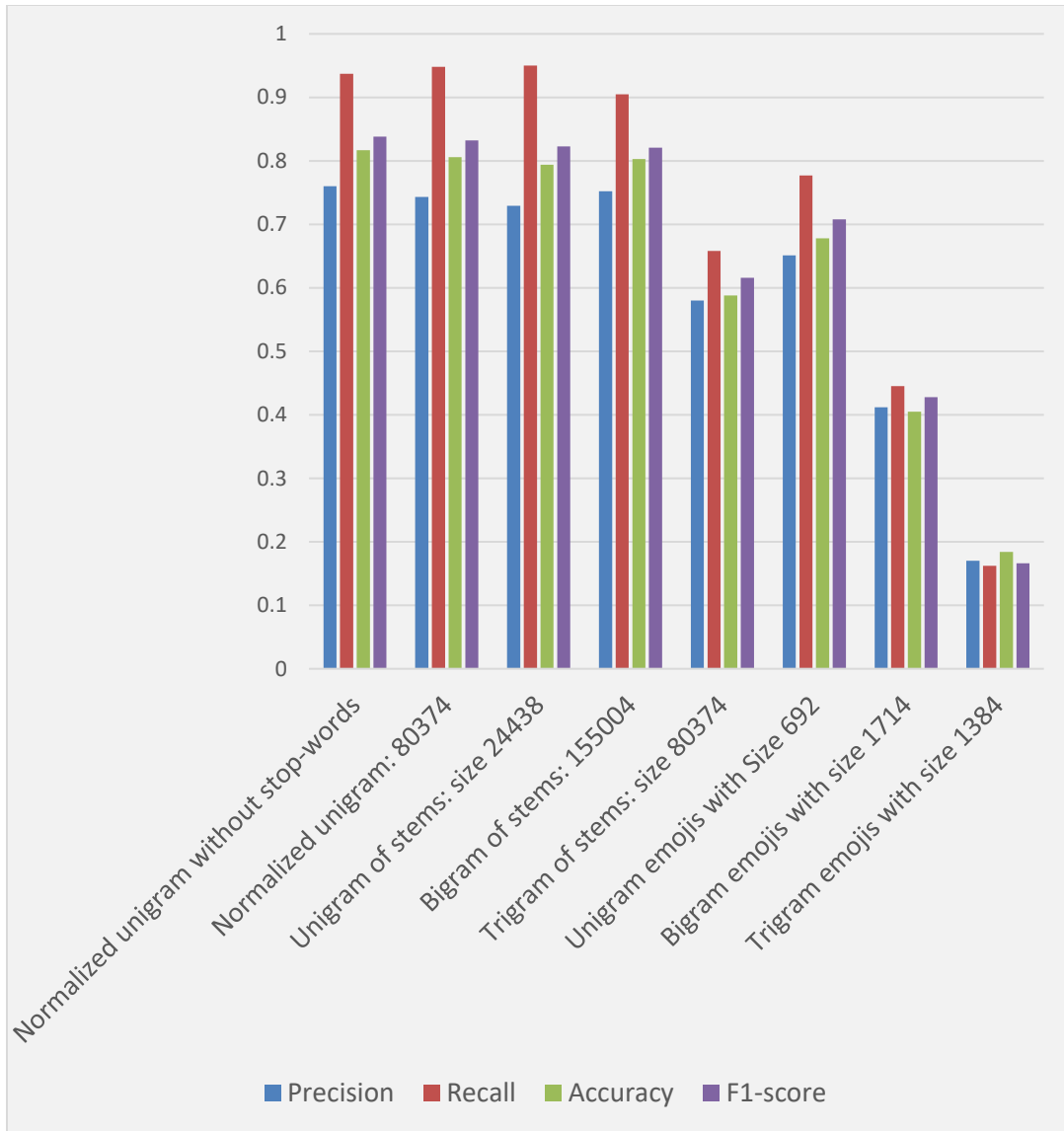


Figure 43. The performance of seven fanatic-lexicons generated by PPMI method

Table 67. The performance of seven fanatic-lexicons generated by the proposed-TFICF method

Token-type	Data type	Precision	Recall	Accuracy	F1-score
Normalized unigrams without stop-words	Text	0.864	0.948	0.899	0.904
Normalized unigrams: size 59373	Text	0.860	0.952	0.897	0.903
Unigrams of stems: size 24438	Text	0.858	0.932	0.888	0.893
Bigrams of stems: size 155004	Text	0.796	0.906	0.836	0.847
Trigrams of stems: size 80374	Text	0.588	0.666	0.597	0.624
Unigrams with size 692	Emojis	0.669	0.757	0.690	0.710
Bigrams with size 1714	Emojis	0.407	0.428	0.402	0.417
Trigrams with size 1384	Emojis	0.168	0.159	0.187	0.163

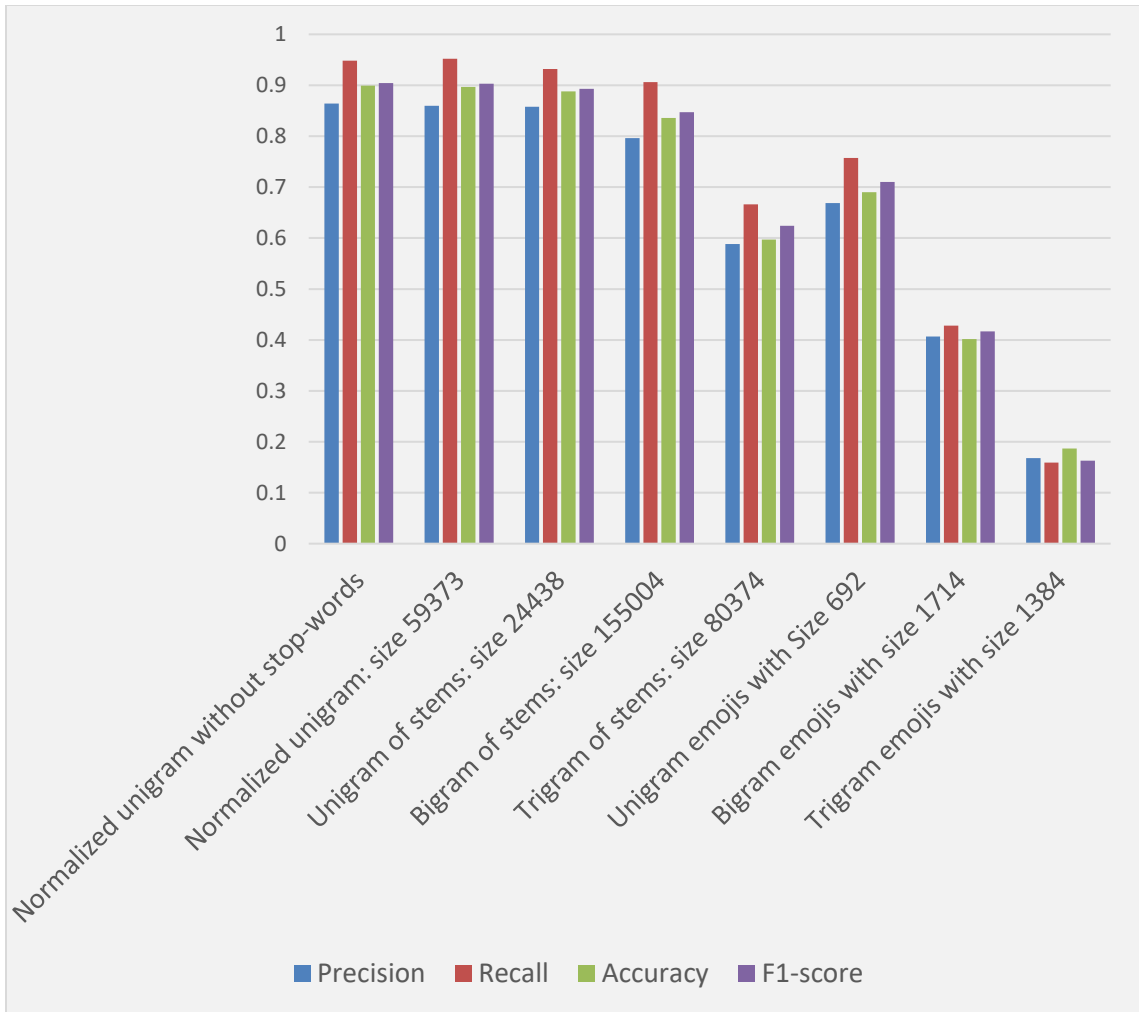


Figure 44. The performance of seven fanatic-lexicons generated by the proposed-TFICF method

9.3 Learning-Based Classification Approach

In this section, the features that were identified and extracted in Chapter 8, were examined for classification. We have examined each feature set a lone and studied the impact of possible combinations. We have five feature sets: stylistic, phonological, morphological, syntactic and semantic. We have used three classifiers: Naïve Bayes, Logistic Regression and Support vector machine. We have considered six cases in this regard. In case 1, we have studied the impact of stylistic features on the problem-domain. In case 2, we have examined the effect of the proposed phonological features. In case 3, the impact of bag-of-

stems and bag-of-emojis features representation was investigated for unigrams, bigrams and both. In case 4, the impact of the transient words on problem-domain was studied. In case 5, we have studied the impact of combining four feature-sets which are: bag-of-stems, bag-of-emojis, bag-of-hashtag, and bag-of-punctuations. In case 6, we have examined the impact of hybrid approach where eight fanatic indicators/features of each tweet were extracted from the proposed lexicons and used as input to the learning-based algorithms and models.

9.3.1 Stylistic Features Impact

In this scenario, six stylistic features are examined which are: character count, word count, unique word count, emojis count, unique emojis count, and average word lengths (See Section 8.1 and Table 63). The purpose of these experiments is to check if there is a relation between the sizes of the elements of the documents and the domain-problem. While we have examined the impact of the document length using the rule-based approach in the previous Section (See Section 9.2), we have found that the low length document performs better than the document with high length. Here, we are examining the impact of six length-related features using learning-based approach to answer the following question: what is the impact of the proposed stylistic feature on the performance?

Table 68 shows the results of using stylistic feature-set where the extracted features were examined using three well-known classifiers which are: LR, NB and SVM. Figure 45 visualizes what are illustrated in

Table 68. We can observe a minor impact of these features with all classifiers. This emphasis that the proposed stylistic features can play role on the domain-problem.

Table 68. The impact of the proposed stylistic features

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	5 folds cross-validation	0.540	0.542	0.541	0.541	0.542
	Test datasets	0.669	0.505	0.604	0.587	0.551
NB	5 folds cross-validation	0.642	0.518	0.592	0.580	0.553
	Test datasets	0.748	0.418	0.624	0.583	0.501
SVM	5 folds cross-validation	0.542	0.611	0.573	0.577	0.591
	Test datasets	0.698	0.485	0.617	0.592	0.543

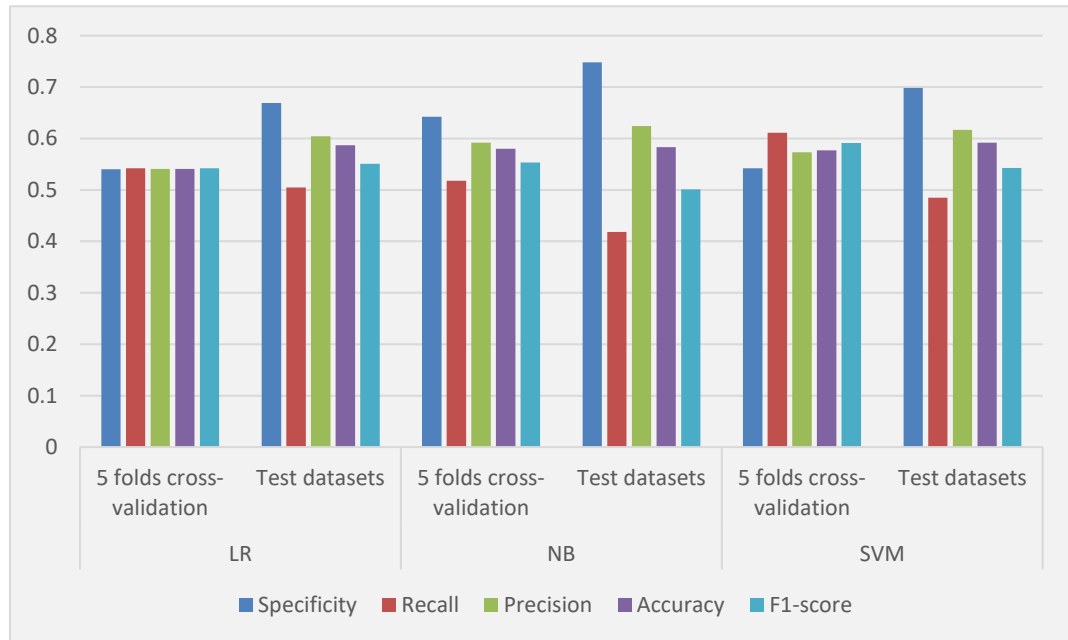


Figure 45. The impact of the proposed stylistic features

9.3.2 Phonological Features Impact

In this scenario, 24 phonological features were examined (See Table 60). The purpose of these experiments are to examine the low-level features and their impact on the domain-problem. We have tried to answer the following question: Is there a relation between the letters related to specific articulation-points, and fanatic-related words? We have pursued some prime analysis in Section 8.1. Table 69 shows the results of examining three classifiers (LR, NB and SVM) built with the proposed 24 phonological features. Figure 46 summarizes what are represented in Table 69. We can observe that there is a minor impact with all classifiers. This emphasizes that phonological features play a light role on the domain-problem.

Table 69. The impact of the proposed phonological features

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.603	0.567	0.587	0.585	0.577
	Separated test dataset	0.598	0.561	0.582	0.579	0.571
NB	90% training & 10% testing	0.760	0.404	0.627	0.583	0.492
	Separated test dataset	0.809	0.341	0.641	0.575	0.445
SVM	90% training & 10% testing	0.607	0.566	0.590	0.587	0.578
	Separated test dataset	0.600	0.562	0.584	0.581	0.573

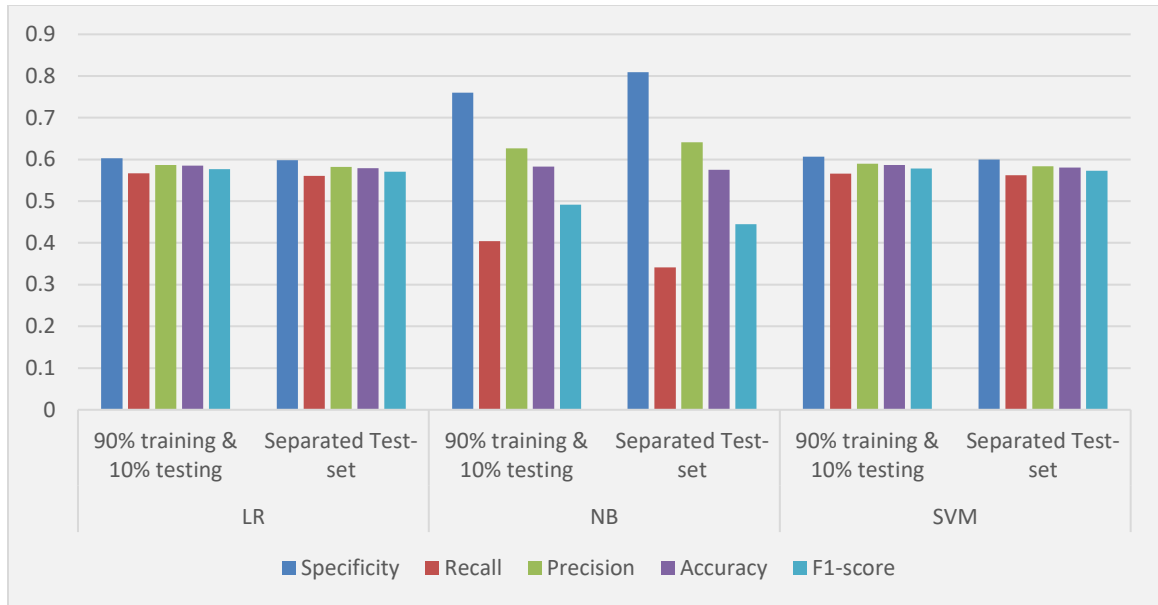


Figure 46. The impact of the proposed phonological features

9.3.3 The Impact of Bag-of-Stems unigrams and bigrams Features

While the previous investigation was applied using letter related features to, this investigation examines features related to word segments. Actually, we address features that are related to bag-of-stems feature-representation. Bag-of-stems representation is a common feature set used in sentiment analysis. In bag-of-word representation, each word in the corpus is converted into a long sparse vector (Refer to Table 35). For this purpose, we have imported our training corpus to Weka. Then, we have converted the tweets into bag-of-words feature representation (using *StringToWordVector Weka* filter). We have selected the binary weighting scheme. We remove low frequency words by selecting 30 as a minimum frequency where a word appears in the corpus less than 30 will be ignored. This process has produced 6265 features (unigrams stems). Next, we have conducted feature selection (using *ChiSquaredAttributeEval in Weka*) to eliminate irrelevant features. The selected features were ranked. After, analyzing the results, we have ignored words with zero rank. Therefore, the highest 3634 features were extracted. The 3634 unique

unigrams words were converted to a feature vector (using *FixedDictionaryStringToWordVector* in *Weka*). We have also filtered stop-words from the feature vector space before classifications. Table 70 shows the classification results of the extracted unigrams bag-of-words feature representation. Figure 47 illustrated what is shown in Table 70. It can be seen that the extracted unigrams bag-of-words features have performed better than stylistic and phonological features. This is because that the word is the main entity where it usually be used to express the feeling. We can also observe that SVM and LR have performed better than NB.

Table 70. Classifications using Unigrams Bag-Of-Stems Feature

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.917	0.909	0.917	0.913	0.913
	Separated test dataset	0.915	0.889	0.913	0.902	0.901
NB	90% training & 10% testing	0.700	0.704	0.704	0.702	0.704
	Separated test dataset	0.770	0.626	0.731	0.698	0.674
SVM	90% training & 10% testing	0.916	0.907	0.916	0.911	0.911
	Separated test dataset	0.914	0.888	0.912	0.901	0.900

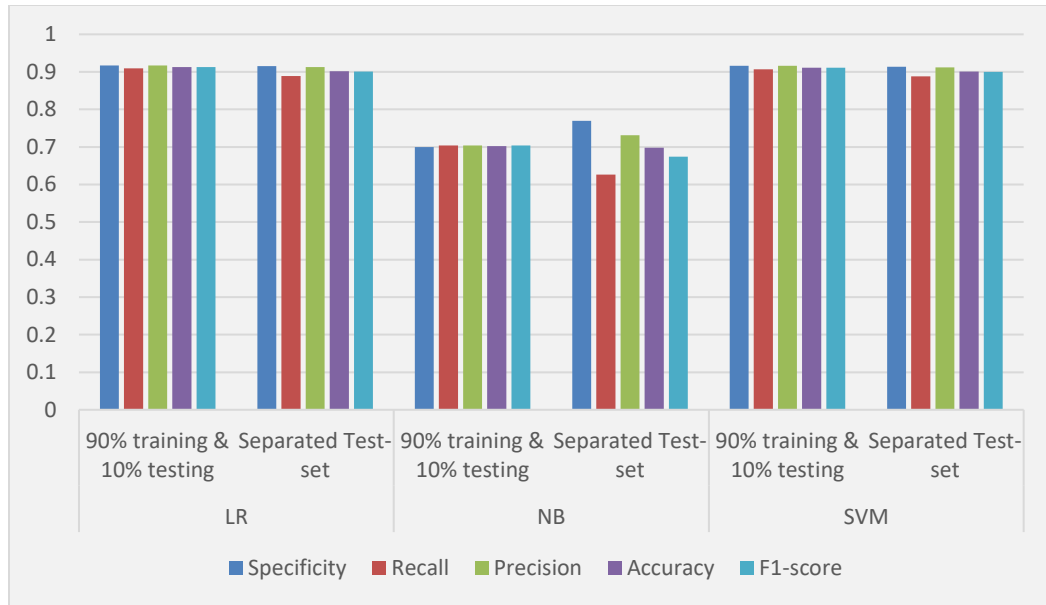


Figure 47. Classifications using Unigrams Bag-Of-Stems Feature

We have repeated the same process for bigrams where we have selected 5315 bigrams phrases (through *ChiSquaredAttributeEval* in *Weka*). Each phrase was represented as a feature in the feature vector space. Table 71 shows the impact of using bigrams bag-of-words feature representation to build fanatic classification model. Three classifiers were examined which are: LR, NB and SVM. Figure 48 illustrates what are represented in Table 71. We can observe that the bigrams features have also achieved better than stylistic and phonological features. We can also notice that unigrams features achieved higher accuracy than bigrams features using LR and SVM. This might be because that the unigrams can cover more tokens in the given text than bigrams. Bigrams bag-of-words feature representation is also important feature because it helps to keep semantic of the ordering of two tokens.

Table 71. Classifications of bigrams bag-of-stems feature

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.795	0.906	0.817	0.851	0.859
	Separated test dataset	0.729	0.850	0.759	0.790	0.802
NB	90% training & 10% testing	0.750	0.735	0.748	0.742	0.741
	Separated test dataset	0.755	0.638	0.723	0.697	0.678
SVM	90% training & 10% testing	0.785	0.910	0.811	0.848	0.858
	Separated test dataset	0.711	0.863	0.749	0.787	0.802

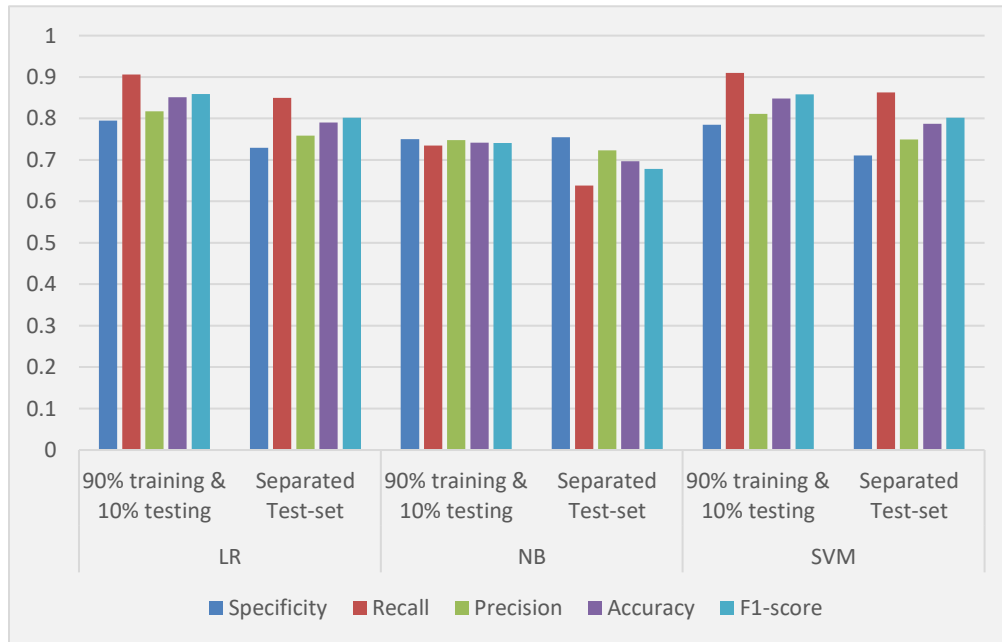


Figure 48. Classification of bigrams bag-of-stems feature

What about combining both unigrams and bigrams together? To answer this question, we have repeated a similar process as previous experiments by combining bigrams and unigrams features. The number of extracted features were 8950 bigrams and unigrams terms. Table 72 shows the results of combining unigrams and bigrams features. Figure 49 visualizes what are presented in Table 72. The main observation is that combining unigrams and bigrams features achieves higher accuracy than using unigrams alone or bigrams alone. This is because that the advantages of both unigrams and bigrams features can be obtained by combining both.

Table 72. Combining unigrams and bigrams bag-of-stems feature Classifications

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.950	0.948	0.950	0.949	0.949
	Separated test dataset	0.903	0.918	0.904	0.910	0.911
NB	90% training & 10% testing	0.742	0.705	0.734	0.723	0.719
	Separated test dataset	0.787	0.629	0.747	0.708	0.683
SVM	90% training & 10% testing	0.945	0.950	0.946	0.948	0.948
	Separated test dataset	0.892	0.921	0.895	0.906	0.908

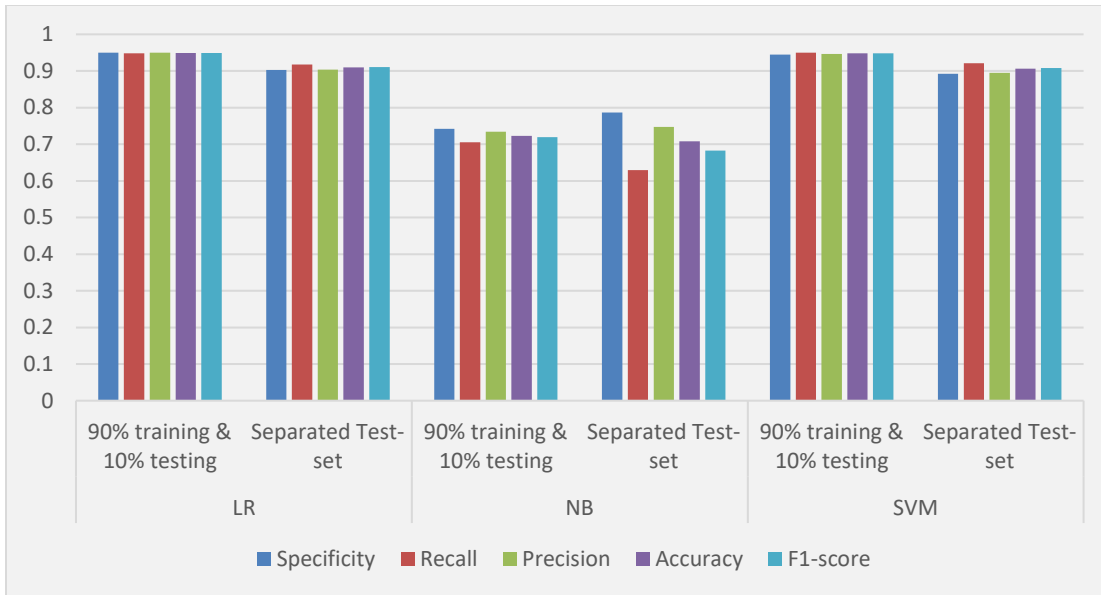


Figure 49. Combining unigrams and bigrams bag-of-stems feature Classification

9.3.4 Emojis Unigrams and Bigrams Impact

In previous section, we have examined the impact of bag-of-words features. In this section, we will examine the impact of using bag-of-emojis features. We will use unigrams and bigrams bag-of-emojis features to build fanatic classification model. Table 73 shows the classification performance using bag-of-emojis feature representation. Three classifiers were examined which are: LR, NB and SVM. Figure 50 illustrates what are shown in Table 73. The results emphasize that bag-of-emojis could play a role on the domain-problem.

Table 73. Combining unigrams and bigrams bag-of-emojis feature Classifications

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.525	0.737	0.609	0.631	0.667
	Separated test dataset	0.591	0.783	0.657	0.687	0.715
NB	90% training & 10% testing	0.840	0.280	0.637	0.560	0.389
	Separated test dataset	0.906	0.247	0.725	0.576	0.368
SVM	90% training & 10% testing	0.526	0.735	0.608	0.630	0.666
	Separated test dataset	0.598	0.785	0.662	0.692	0.718

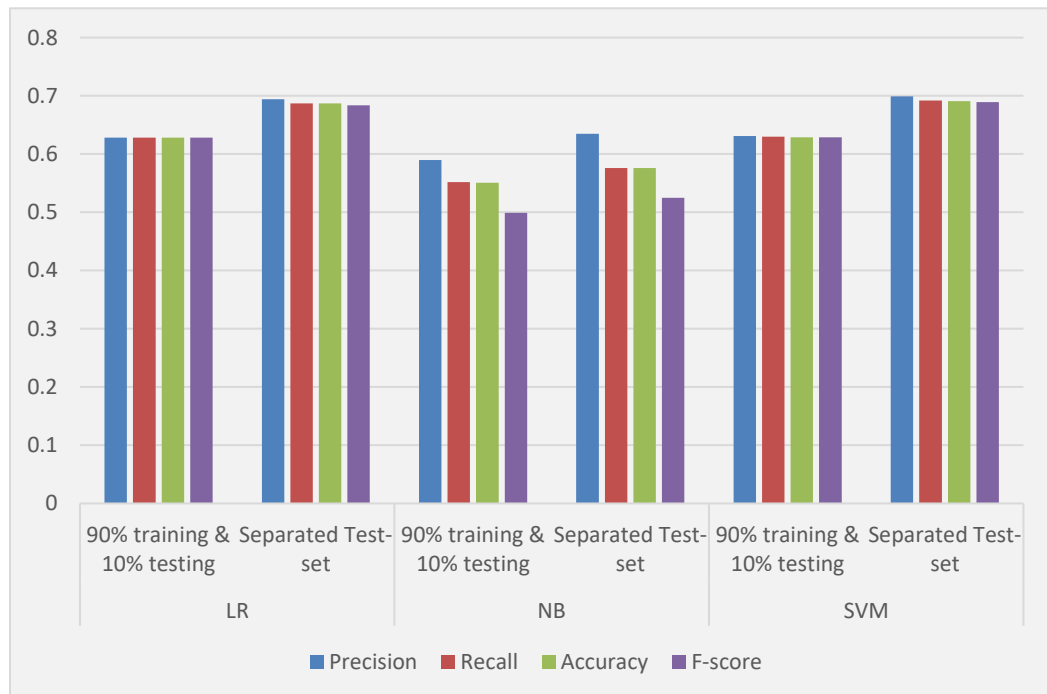


Figure 50. Combining unigrams and bigrams bag-of-emojis feature Classifications

9.3.5 Punctuation, hashtag, emojis and stems

Table 74 shows the results of combining four feature-sets which are: bag-of-punctuations, bag-of-hashtag, unigrams and bigrams of bag-of-emojis, and unigrams and bigrams bag-of-stems. Chi-Square feature selection technique (ChiSquaredAttributeEval in Weka) has been applied to select relevant features. The total number of selected features are 9593 where 432 emojis, 8950 stems, 10 punctuations and 211 hashtags. Figure 51 reflects what are presented in Table 74. We can observe that combining those features lead to higher performance than use each feature-set alone. This is because that more information about the domain-problem is added.

Table 74. Combining punctuation, hashtag, emojis and stems features Classifications

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.949	0.949	0.950	0.949	0.949
	Separated test dataset	0.902	0.920	0.904	0.911	0.912
NB	90% training & 10% testing	0.741	0.708	0.735	0.724	0.721
	Separated test dataset	0.787	0.631	0.747	0.709	0.684
SVM	90% training & 10% testing	0.946	0.948	0.946	0.947	0.947
	Separated test dataset	0.893	0.921	0.896	0.907	0.908

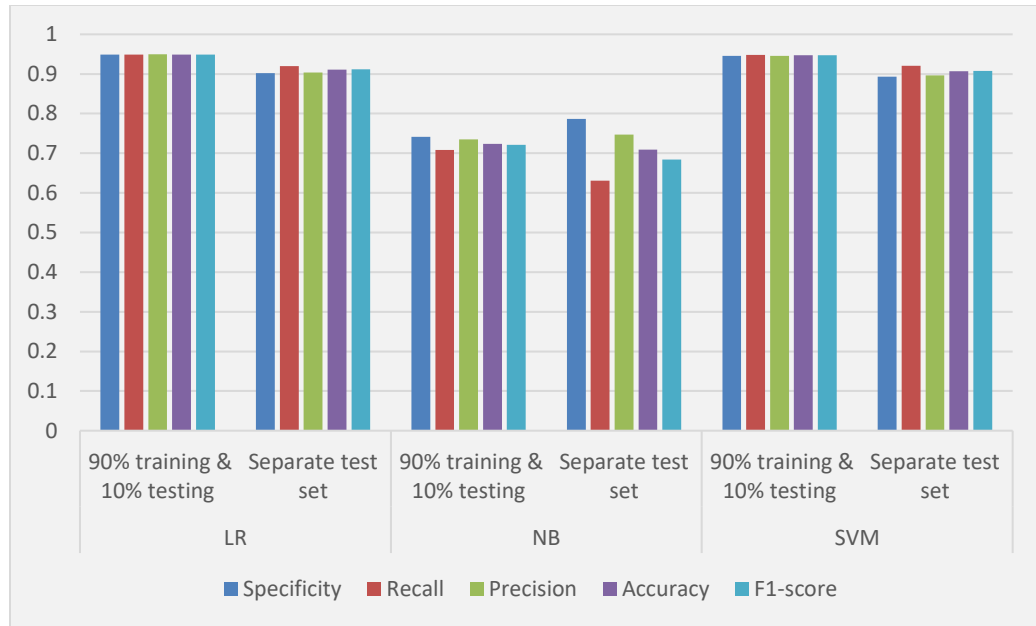


Figure 51. Combining Punctuation, hashtag, emojis and stems features Classifications

9.3.6 Syntactical Features

In this scenario, twenty-six syntactical features are examined (See Sub-section 8.1.3 and Table 61). The purpose of these experiments is to examine the impact of different types of transient words on the domain-problem. Table 75 shows the impact of using the proposed syntactical features to build fanatic classification model. Figure 52 replicates what are displayed in Table 75. In this section, we try to answer the following question: Is there a relation between different types of transient words and the domain-problem? Two weighting schemas were extracted which are: count and ratio. We proposed around 26 group of transient words which are: intensifier words, modal words, negation words, etc. Each group represents one feature. For example, we extracted the count of the intensifier words (e.g. جداً، تماماً، للغاية) in a document as one feature. We can observe that F1-score matrix represents a positive impact with all classifiers. The results emphasize that syntactical features can play a role on the domain-problem.

Table 75. The impact of the proposed syntactical features

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.530	0.620	0.568	0.575	0.593
	Separated test dataset	0.606	0.549	0.582	0.578	0.565
NB	90% training & 10% testing	0.415	0.726	0.553	0.571	0.628
	Separated test dataset	0.466	0.663	0.554	0.565	0.604
SVM	90% training & 10% testing	0.489	0.687	0.572	0.588	0.624
	Separated test dataset	0.607	0.547	0.582	0.577	0.564

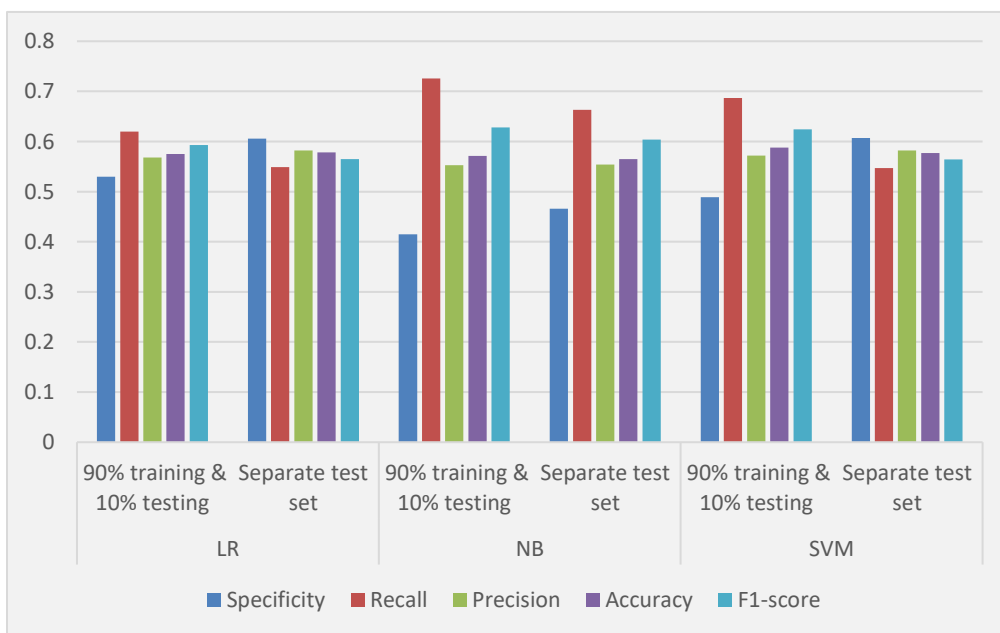


Figure 52. The impact of the proposed syntactical features

What about representing trainset words as a bag-of-words feature representation? Table 76 shows the impact of this feature representation which contains 363 transient words where each word represented as one feature. Each feature is represented as a long sparse vector in the feature vector space. Figure 53 reflects what are presented in Table 76. From Table 75 and Table 76, we can observe that the features with the bag of transient words perform better than the dans features. This might because that each group contains fanatic and non-fanatic related-words or each word in one group can has different impact on the domain-problem. For example, in the model words feature used to generate the result of Table 75, all these words (يجب، لازم، ممكن، ضروري) are represented as one feature where we can notice that the words (ضروري، يجب، لازم) are more related to fanatic class and the word (ممكن) is more related to non-fanatic class. This means that combining such words in one group to represent one feature will remove some information that is important to the domain-problem which lead to reduce the classification-performance.

Table 76. The impact of transient words as bag-of-words feature representation

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	90% training & 10% testing	0.456	0.776	0.588	0.616	0.669
	Separated test dataset	0.520	0.742	0.607	0.631	0.668
NB	90% training & 10% testing	0.411	0.765	0.565	0.588	0.650
	Separated test dataset	0.468	0.716	0.574	0.592	0.637
SVM	90% training & 10% testing	0.448	0.783	0.587	0.616	0.671
	Separated test dataset	0.514	0.748	0.606	0.631	0.670

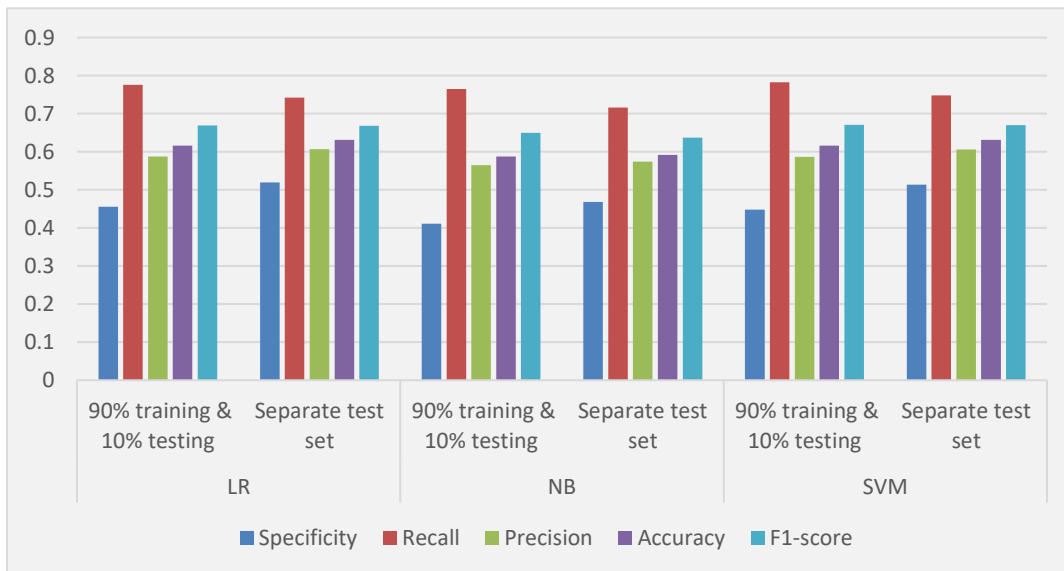


Figure 53. Applying Bag-of-Words Syntactical features

9.4 Hybrid Classification Approach

In this approach, text unigrams and bigrams lexicons that have been generated by PPMI and TFICF were used to extract features as input for learning-based models as shown in Figure 54.

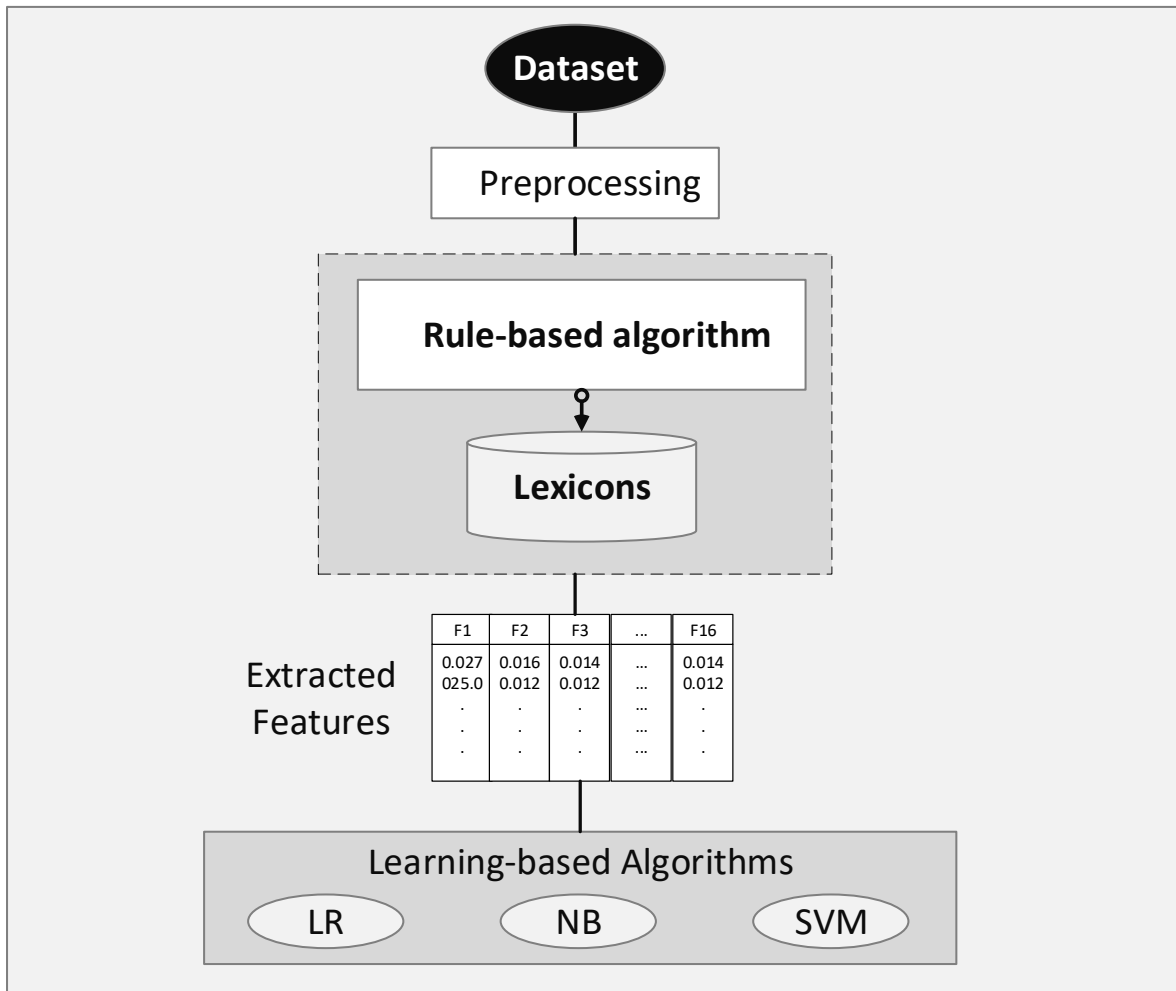


Figure 54. Hybrid classification approach

The scores for eight fanatic-indicators/features (discussed in Section 4.1) were calculated from each lexicon (unigrams and bigrams lexicons). This process has produced two groups with 16 features. Then, the eight unigrams features were normalized by row-wise

normalization where the score of each feature is divided by the total sum of the eight features. Normalization was also applied on the second group that has been generated by bigrams lexicon. As a result, our training and testing data have 16 normalized features. We have built a java program to extract the score of each feature for each tweet in the lexicons. To apply learning-based classifiers, we have utilized the same classifiers that have been used in previous scenarios with the same configurations. We have extracted 32 features from the proposed-TFICF lexicons. The purpose of applying this approach is to compare the performance of the extracted features from each lexicon type (PPMI and TFICF). This process has answered the following questions: Does hybrid approach perform better than learning-based approach, or rule-based approach? Do the extracted features perform better than a bag-of-words feature representation? Which lexicon-type would perform better (PPMI or the proposed-TFICF)? Table 77 shows the results of the classification using the hybrid approach with the extracted 16 features using PPMI lexicons. Figure 55 visualizes what are presented in Table 77. We can observe that this feature type and approach achieve high performance. This is because that hybrid approach gains the advantages of the previous two approaches which are: rule-based approach and learning-based approach. We can also observe that the NB classifier performs better than SVM and LR classifiers which is not the case with previous experiments. This might be because several factors such as the type of the features where the input features in the previous experiments are sparse features. In the other hand, the input features of the hybrid approach are dense features. Another factor might be the number of the input features to the classifiers.

Table 77. Classifications using the proposed 16 features extracted from PPMI lexicons

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	10-fold cross validation	0.884	0.865	0.882	0.875	0.873
	Separated test dataset	0.876	0.797	0.865	0.837	0.830
NB	10-fold cross validation	0.909	0.849	0.904	0.879	0.875
	Separated test dataset	0.911	0.811	0.900	0.861	0.853
SVM	10-fold cross validation	0.886	0.863	0.884	0.874	0.873
	Separated test dataset	0.879	0.796	0.867	0.838	0.830

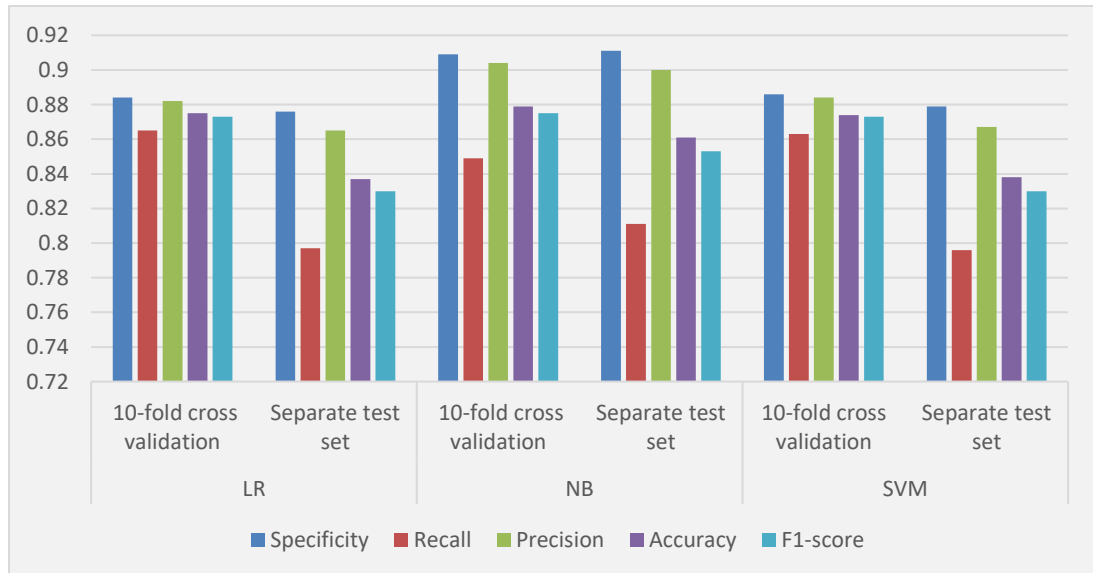


Figure 55. Classifications using the proposed 16 features extracted from PPMI lexicons

Table 78 shows the results of the classification using hybrid approach with the extracted 16 features from the proposed-TFICF lexicons. Figure 56 illustrates what are shown in Table 78. It can be seen that the results of the extracted features using the proposed-TFICF (see Table 78) outperforms the extracted features using PPMI (see Table 77). This observation introduces another evidence about the usefulness of the proposed-TFICF method to generate the multi-sentiment lexicons automatically.

Table 78. T Classifications using the proposed 16 features extracted from TFICF lexicons

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	10-fold cross validation	0.927	0.916	0.926	0.922	0.921
	Separated test dataset	0.902	0.880	0.899	0.891	0.889
NB	10-fold cross validation	0.944	0.903	0.942	0.924	0.922
	Separated test dataset	0.915	0.880	0.911	0.897	0.895
SVM	10-fold cross validation	0.928	0.915	0.927	0.921	0.921
	Separated test dataset	0.904	0.880	0.901	0.892	0.890

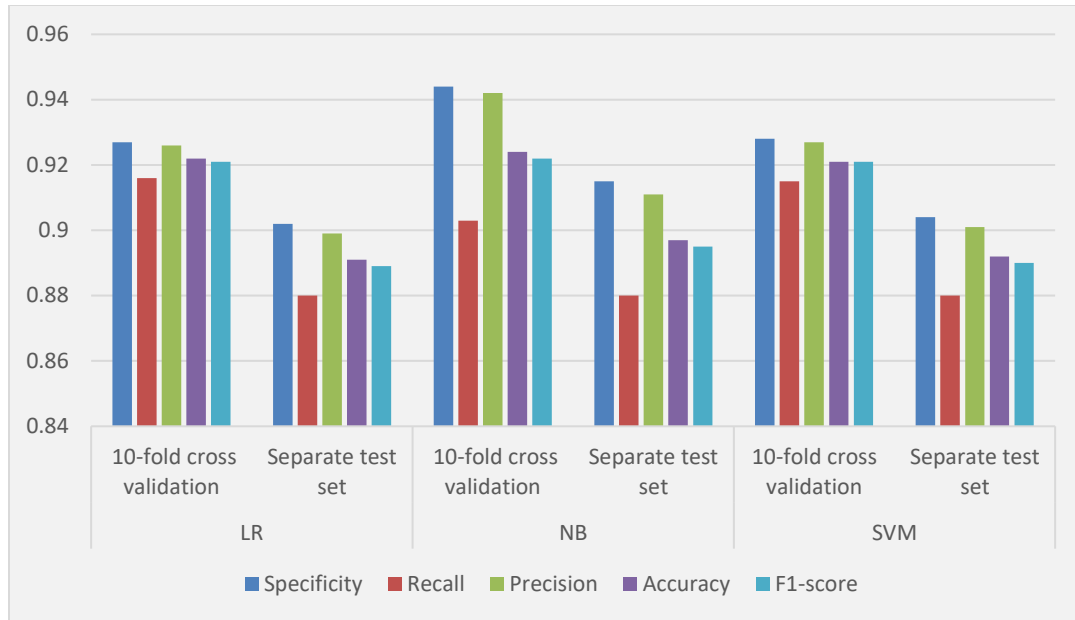


Figure 56. Classifications using the proposed 16 features extracted from TFICF lexicons

What about if we conduct classification using hybrid approach that combines the extracted 16 features from PPMI lexicons, and the extracted 16 features from TFICF lexicons? Table 79 shows the results of the classification using the combination of the 32 features that have been extracted from both lexicons TFICF and PPMI. Figure 57 illustrates what are displayed in Table 79.

In Table 77, Unigrams and bigrams Lexicons generated by PPMI method were used to the extract 16 features. In Table 78, Unigrams and bigrams Lexicons generated by proposed-TFICF method were used to the extract 16 features. Clearly in Table 79, we can observe higher performance by combining the features used to generate the results presented in Table 77, and the features used to generate the results presented in Table 78.

Table 79. Classifications using the proposed 32 features extracted from both PPMI and TFICF lexicons

Classifiers	Test datasets	Specificity	Recall	Precision	Accuracy	F1-score
LR	10-fold cross validation	0.938	0.928	0.938	0.933	0.933
	Separated test dataset	0.892	0.890	0.891	0.891	0.891
NB	10-fold cross validation	0.936	0.886	0.933	0.911	0.909
	Separated test dataset	0.923	0.864	0.918	0.894	0.890
SVM	10-fold cross validation	0.939	0.928	0.938	0.934	0.933
	Separated test dataset	0.887	0.890	0.887	0.889	0.888

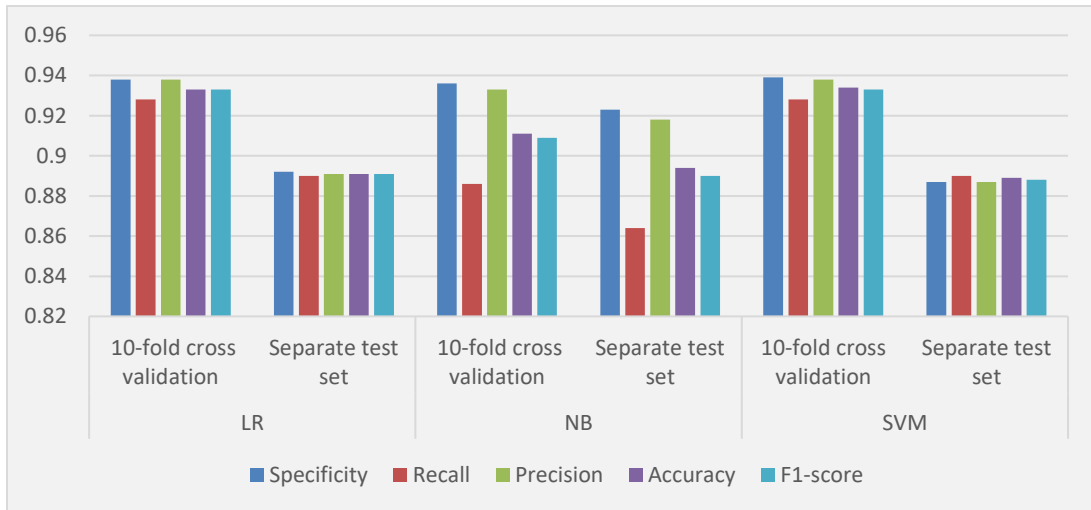


Figure 57. Classifications using the proposed 32 features extracted from both PPMI and TFICF lexicons

9.5 Summary

In this chapter, we examined three classification approaches which are rule-based, learning-based and hybrid. The results showed that the learning-based performs slightly better than rule-based approach. This is because the rule-based uses labeled senti-lexicons to classify the given document where the senti-lexicons maintain only the semantic of word-to-words association and word-to-polarity association. On the other hand, the learning-based algorithms “understand” more details where they learn the semantic of the word-to-words association, word-to-documents association, word-to-polarities association and document-to-polarities association. We observe that hybrid approach performs better than rule-based approach because the hybrid approach adds new two dimensions of information to the semantic of rule-based approach which are context-to-documents association and context-to-polarities association. However, the results showed that some experiments of learning-based perform a little bit better than hybrid approach. This might be because that the hybrid approach loses some information of word-to-document association and word-to-polarities association when this information was encapsulated in context-to-documents association and context-to-polarities association respectively. However, hybrid approach has been applied with a very low number of features (32 features) compared to learning-based approach (9000 features). Nevertheless, hybrid approach still achieves high accuracy, recall, precision and F1-score. This means that hybrid approach is more applicable to apply and examine using more machine learning algorithms.

For classifiers, we observe that LR and SVM performed better than NB on all learning-based experiments. On the other hand, we observe that NB performs better than SVM and

NB on hybrid-based experiments. Two factors are behind this observation. First, the type of the features in the learning-based approach is vectorized into long sparse vectors, and the dataset in hybrid-based approach is vectorized into long dense vectors. Second, the number of features in learning-based approach is huge, and the number of features in hybrid-based approach is small (16 or 32 features). We can observe that the performance of LR and SVM is close. This is because that the SVM is a kind of LR with confidence margin.

For the used types of features, we observe that using the following feature alone lead to poor performance: stylistic, phonological and syntactical feature-sets. This is because these feature-sets usually do not have a stable pattern. For example, in the stylistic, the high-length document may be considered as non-fanatic by assuming that the writer writes a long document because he/she is trying to justify his/her opinions. Providing justification in the document is more related to non-fanatic, but this is not always the case where the fanatic also can appear in long documents. Sometimes, a long document may contain fanatic and non-fanatic opinions. For this reason, such feature might have good impact when combined with other features. We can observe that combining hashtags, punctuation, emojis and stems feature-sets lead to high performance. This is because that more information about the domain-problem is added. Combining these four feature-sets will obviously lead to better results. By analyzing the four feature-sets, we can observe that hashtag usually contains important keywords about a document. Emojis usually used to represent emotion related to the domain-problem which can clearly have important impact on this work. The bag-of-stems is an important feature by natural for sentiment analysis

where the words are the means of expressing feelings in text-classification. Punctuations feature-set can also have some information about the domain problem.

We can notice that the bag-of-words feature representation of unigrams and bigrams achieved higher performance than unigrams alone or bigrams alone. This is because that unigrams alone do not keep the information of words' ordering while the bigrams alone results in low coverage where the probability of two words appear together in the given document is lower than one words appear in the document. Consequently, combining both unigrams and bigrams will obtain the advantages of both where we can have high coverage and maintain the semantic of the order of two words.

We observe that the performance of the proposed TFICF method outperforms PPMI method in all classification experiments. The reason for that might be because PPMI replaces some information with zero and this information might contain some related semantic to domain-problem.

CHAPTER 10

DISCUSSION AND CONCLUSION

Sports-fanaticism is an issue that requires more attention. While social media provides a communication tool for fans, we recommend Anti-fanatic application and tools that work like anti-virus software to detect the fanatic-text automatically and provide measurement tools for fanatic-text. As far as we know, this is the first study that formalizes this concept to build classification models that can automatically detect fanaticism in text.

Building resources for Arabic sentiment analysis is valuable but time and effort consuming. During this study, we found that there is a lack of domain-specific lexicons. Specially, there is a lack of labeled corpora and senti-lexicons in sports domain for Arabic sentiment Analysis. Twitter is an important data source for sentiment analysis. The size of a tweet is closed to the size of a sentence, so the probability that the content of the tweet has only one opinion is high. Moreover, Twitter is rich of dialects where the generated lexicons and corpora from tweets can be applied in real-life application to analyze the sentiments. Statistical methods used in sentiment analysis depend on the co-occurring of words in the positive and negative documents. We have used Mongoddb database during building resources for Arabic sentiment analysis. Using Mongoddb database has two advantages. First, it provides scalable functionality to add new attributes to data without caring about the schema. For example, the original text of tweets can be cleaned, and the result of cleaning can be added to new fields. Second, it can deal with big-data and provide easy engines and APIs to access and retrieve the data.

We have found that the preprocessing of Arabic text is crucial in Arabic sentiment analysis as it helps in feature-reduction process. While root stemmers lead to over-preprocessing, the segmentation and the normalization are not enough as they can lead to under-preprocessing. Selecting a suitable stemmer is essential. There is a need for a stemmer that can also handle dialects because social media is rich of informal text

Most of sentiment analysis studies have applied bag-of-words feature representation as it provides direct indicators for sentiment analysis. However, there are other features that can help to analyze sentiment such as emojis, stylistic features and others. While the bag-of-words feature representation of unigrams does not keep the semantic of the words' orders, we recommend that using unigrams and bigrams. Using bigrams can keep the semantic of the order of word pairs while unigrams provide high coverage. Our result showed that using both can improve the classification performance.

10.1 Conclusion

In this study, sports-fanaticism is formalized into indicators with the purpose of building corpora, lexicons and classification models. Sports-fanaticism in Arabic social text is an issue that required Anti-fanatic tools, application and software. This study provides foundation to automatically detect sports-fanaticism in text. While the fanatic is related to sentiment, sentiment analysis techniques were applied. Two corpora were collected and annotated. Web application for annotation was developed. As a result, 10K tweets was validated manually. On the other hand, 300K tweets were annotated automatically using keywords. Three fanatic-lexicons were built. One was built manually which contains around 2K phrases while the other two were built automatically using PPMI and the

proposed-TFICF. Three tasks were applied for preprocessing which are: cleaning, normalization and stemming. During this study, java tools for annotation and preprocessing were implemented. Five feature-sets were extracted and examined which are: stylistics features, phonological features, morphological feature, syntactical feature and semantic features. For classification, NB, LR, and SVM were applied. We have achieved 94% accuracy using LR with semantic features.

As far as we know, we have proposed the first sport-fanaticism formalism that formalize the concept of sports-fanaticism into criteria, indicators, features and definitions. The proposed formalism can be used as a guide for building corpora, lexicons and classification models in the context of sentiment analysis and machine learning. Apparently, there are no proposed sentiment resources that help conducting sentiment analysis to automatically detect sport-fanaticism. Therefore, we have built the first fanatic-lexicons and annotated corpora for sport-fanaticism detection. We have also built the first classification models for sport-fanaticism detection. In addition, we have proposed new method to automatically generate multi-classes senti-lexicons. We have called our proposed method as “Term-Frequency Inverse-Context Frequency (TFICF)”. The proposed method has shown better performance than Positive Pointwise Mutual Information (PPMI) method. Moreover, we have proposed new types of features that have shown a positive impact on sentiment analysis.

10.2 Developed Tools

This section lists some of the developed tools during this research work.

10.2.1 A Fanatic/ Non-Fanatic Classifier

For prove of concept, we have developed a web-tool to classify a given text into Fanatic and non-Fanatic. In the tool, we have used the unigrams bag-of-stems feature representation classification model discussed in Section 9.3.3. It works as follows: First, the given text is preprocessed as discuss in Section 6.1 and Section 6.2. Then, the preprocessed text is stemmed using Farasa stemmer (See Section 6.3). Then, the proposed classification model is applied to determine the predicted class of the given text. In addition, the tool calculates the score of the given text by summing all the weights (provided by the model) of all the words of the given text. Figure 58 shows a screenshot of the developed tool.

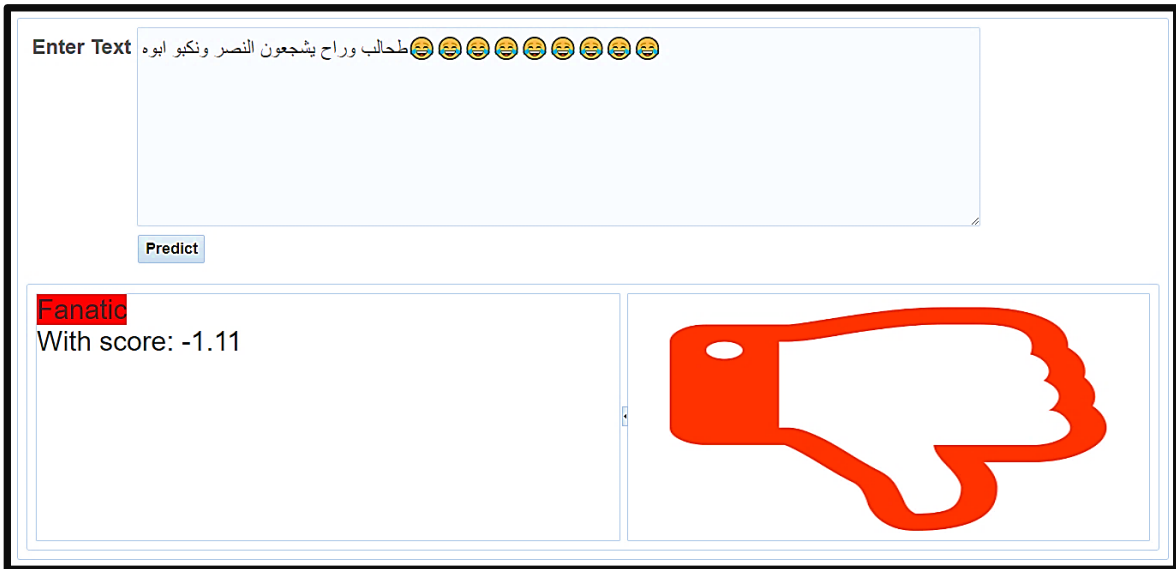


Figure 58. Fanatic Classification Tool

10.2.2 An Annotation Tool

We have also developed an annotation tool to simplify the process of tweets labeling. The tool was developed using Oracle ADF, Java and MYSQL. The tool provides information about each tweet where each tweet has two labels: negative (fanatic) or positive (non-

fanatic). Each tweet also has a reason field that shows why such label is determined as fanatic (negative) or non-fanatic (positive). This video (<https://youtu.be/9CQaVTRA6D8>) shows how the process of annotation using the developed tool was conducted. Figure 59 shows a screenshot of the developed annotation tool.



Figure 59. Annotation Tool

10.2.3 A Cleaner

We have developed a cleaner tool for preprocessing tasks such as removing non-Arabic characters, removing URL, removing digits, removing punctuations, removing duplicate spaces, removing lengthening characters, etc. The cleaner is a java program developed mainly for Arabic sentiment analysis. Cleaning is necessary in sentiment analysis where the collected data has usually noise and irrelevant entities.

10.2.4 A Normalizer

We have utilized, modified and adapted IBM normalizer [119] to accomplish the required normalization tasks such as removing Arabic diacritics (tashkeel), removing tatweel, normalizing shapes of Alef, normalizing tah marbouta to hah, normalizing yah maksoura

to alef maksoura, normalizing ben, and normalizing Abdal. Table 37 shows examples for these types of normalization. The adapted normalizer has been developed using Java programming language.

10.2.5 A Feature Extraction Tool

We have implemented a tool to extract the proposed features (See Chapter 8). We used Java programming language to implement the tool.

10.2.6 Other Tools

We have implemented the PPMI and the proposed-TFICF methods discuss in Section 7.2. This tool has been implemented using python. The implemented programs help to generate lexicons automatically from the labeled corpora. We have also implemented tools to simplify conducting classification using rule-based and hybrid-based approaches. Java programming language was used to implement these tools.

10.3 Limitations and Future Work

Although the collected dataset contains around 300K unique tweets for training and around 10K for testing, the dataset is limited to tweets that were collected at short time period between October 22, 2018 and December 16, 2018. We believe that collecting big data posted in three or four years can help to generate more useful resources under consideration. A future work can include conducting a big-data technique to collect and handle big-data.

The generated classification model is limited to tweets of Twitter with limited size. Therefore, using this model for classifying documents with big sizes may have poor

prediction. The reason is that a big document can have many opinions toward different entities while a tweet size has high probability to encapsulate one opinion toward one entity (closed to sentence-level). Our analysis showed that a short document performs better than a big document. This issue can be handles as a future work by applying aspect-level sentiment analysis.

Our work is limited to 3 shallow-learning classifiers (SVM, LR and NB). However, these three classifiers were selected by analyzing the literature where they have shown high performance. A future work might examine the performance of other classifiers and study the impact of deep-learning techniques.

This work is also limited to binary-classifications. However, the built lexicons and corpora were built to be scalable to conduct multi-classifications. The generated lexicons have scores for eight indicators which can be scaled to conduct multi-classifications for eight classes or less. Also, the corpora have information about eight-indicators and 22-contexts which can be scaled to conduct multi-classifications. As a future work, multi-classifications could be conducted to classify a give text into our eight formalized categories (aggression, agitation, hatred, passion, adaption, tolerance, respect or knowledge).

The manual annotation process for test dataset is limited to the available resources (e.g. one annotator). However, the proposed framework is useful for the same situation to generate representative test dataset and can be generalized. The proposed framework considers the following concerns: bias, redundancy, coverage, subjectivity and balance.

To avoid bias, we have collected a large-enough test dataset separately at a different time period from the collected train dataset. The collected test dataset has around 50K unique

tweets. Those 50K tweets was filtered to minimize the redundancy and maximized the coverage to assure that our test dataset covers most of the cases required to measure the classification models. We also need to assure that test dataset is not biased to specific observations, contexts, indicators and keywords in our formalism. We have also examined four under-sampling techniques for balancing in order to assure that the test dataset is not biased to a specific orientation. The subjectivity of annotation was handled by automatically annotating the given corpus using seed-keywords guided by the proposed formalism. Then, the human annotator has validated and checked if there are any incorrect annotated tweets that have caused by one or more of the following: negation (e.g. بلا طموح), two opposite opinions (e.g. انت انسان تافه، معا احترامي،) where one dominates the other, ambiguate of some words where some words can be expressed in fanatic-contexts (e.g. hate/بغض) and non-fanatic contexts (e.g. no matter/بغض النظر), and/or other situation that can be recognized by a human annotator. Such cases may result in flipping the label from positive to negative or vice versa. The developed system for annotation process provides useful information about each tweet where each tweet has the label and the reason field that shows why such label is determined as fanatic or non-fanatic. This video (<https://youtu.be/9CQaVTRA6D8>) shows you the process of annotation using the developed system.

Getting help of more human expert annotators seems to be need as a future expansion of this research work.

References

- [1] B. Liu, *Sentiment Analysis and Opinion Mining*, Morgan and Claypool Publishers, 2012.
- [2] N. Al-Twairsh, H. Al-Khalifa and A. Al-Salman, "Subjectivity and sentiment analysis of Arabic: trends and challenges," in *IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, Doha, Qatar, 2014.
- [3] M. El-Masri, N. Altrabsheh and H. Mansour, "Successes and challenges of Arabic sentiment analysis research: a literature review," *Social Network Analysis and Mining- Springer*, vol. 1, no. 7, p. 54, 2017.
- [4] ف. ب. ع. ع. ا. الطيار, "التعصب الرياضي وأثره في الروابط الأسرية : دراسة ميدانية من وجهة نظر مجلة البحوث الأمنية (السعودية)", *معلمي التربية البدنية ورواد النشاط بالمدارس الثانوية*, vol. 025, no. 063, 2016.
- [5] مركز الملك عبدالعزيز "محمد بن علي العتيق", "التعصب الرياضي أسبابه وأثاره وسبل معالجته بالحوار", *للحوار الوطني*, no. 15, 2013.
- [6] جامعة نايف العربية للعلوم الأمنية "ر. ا. السلمي", "التعصب الرياضي وتأثير وسائل الإعلام الجديدة", pp. 445-486, 2014.
- [7] "ع. ا. ب. ح. الثنيان", "شبكات التواصل الاجتماعي والتعصب الرياضي and أ. موكل, ر. ا. السلمي", *جامعة نايف العربية للعلوم الأمنية*, pp. 487-505, 2014.
- [8] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, "Experimentation in Software Engineering," in *Steps in the Experiment Process*, Springer Science & Business Media, 2012, pp. 85-86.
- [9] S. AlOtaibi and M. B. Khan, "Sentiment Analysis Challenges of Informal Arabic Language," *International Journal Of Advanced Computer Science And Applications - Science & Information*, vol. 2, no. 8, pp. 278-284, 2017.
- [10] T. Al-Mosmi, M. Albared, A. Al-Shabi, N. Omar and S. Abdullah, "Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis," *Journal of Information Science - SAGE Publications Sage UK: London, England*, vol. 0, no. 0, 2017.
- [11] N. Boudad, R. Faizi, R. O. H. Thami and R. Chiheb, "Sentiment analysis in Arabic: A review of the literature," *Ain Shams Engineering Journal - Elsevier*, p. 12, 2017.
- [12] ر. ا. السلمي, "التعصب الرياضي وتأثير وسائل الإعلام الجديدة", "تأليف الندوة العلمية حول دور الإعلام الرياضي في الحد من التعصب والعنف في الملاعب- جامعة نايف العربية للعلوم الأمنية، الرياض، 2014.
- [13] A. Alshehri, C. Don Belcher, S. Colclough and W. Harris, *The effect of increasing awareness about the use of social media on Sport fanaticism for Saudi soccer fans*, Middle Tennessee State University, 2016.
- [14] L. Castillo, D. Brossart, C. Reyes, C. Conoley and M. Phoummarath, "The influence of multicultural training on perceived multicultural counseling competencies and

implicit racial prejudice," *Journal of Multicultural Counseling and Development*, vol. 4, no. 35, p. 243-249, 2007.

- [15] Y.-B. Elisabeth, *The anatomy of prejudices*, Harvard University Press, 1998.
- [16] ع. ب. غ. الشهري و ح. ب. ع. الربعان، التعصب الرياضي ومدى علاقته ببعض نظم التنشئة الاجتماعية [16] لدى الجماهير السعودية في مدينة الرياض، الرياض: جامعة الملك سعود، كلية التربية البدنية، 2009.
- [17] *المجلة العربية للدراسات*، "إ. أ. م. إبراهيم، "أسباب التعصب الجماهيري الرياضي وعلاجه : رؤية شرعي" vol. 031, no. 064, 2016.
- [18] ج. ف. العرجان، "الأدوار الإيجابية والسلبية لوسائل الإعلام الرياضية الأردنية في مستوى العنف والشغب جامعة نايف العربية للعلوم الأمنية-الندوة العلمية حول دور "، والتعصب في منافسات كرة القدم الأردنية 2014، *الإعلام الرياضي في الحد من التعصب والعنف في الملاعب*.
- [19] *اشغال المؤتمر الدولي : الإعلام*، "ع. معزیز، "الإعلام الرياضي ودوره في مواجهة التعصب الرياضي ورهان التنمية - مختبر التواصل وتقنيات التعبير - كلية الآداب والعلوم الإنسانية ظهر المهرار بفاس - المغرب" vol. 000, no. 000, 2016.
- [20] "أ. ط. س. محمد، "الإعلام الرياضي المرئي و دوره في نبذ التعصب و نشر ثقافة الروح الرياضية المؤتمر العلمي العربي الثامن : الإنتاج العلمي التربوي في البيئة العربية - القيمة والأثر - جمعية الثقافة من مصر" vol. 000, no. 000, 2014.
- [21] ب. أ. طامع، "الإعلام الرياضي ودوره في الحد من ظاهرة التعصب وشغب الجماهير في الملاعب" *جامعة نايف العربية للعلوم الأمنية*، "الفلستينية" pp. 325-344, 2014.
- [22] ع. ا. ب. ح. الثنيان، "المواقع الإلكترونية للأندية السعودية كوسيلة إعلامية لمواجهة التعصب الرياضي" *جامعة نايف العربية للعلوم الأمنية*، "والثقافة الأمنية" pp. 413-444, 2014.
- [23] *مجلة الابداع*، "م. خالد، "أهمية تكنولوجيا الاعلام والاتصال في الحد من ظاهرة التعصب الرياضي" *جامعة المسيلة الرياضي*، no. 008, pp. 94-115, 2012.
- [24] م. ع. ب. سعيد، "تقرير حول ندوة : دور الإعلام الرياضي في الحد من التعصب والعنف في الملاعب" *المجلة العربية للدراسات الامنية والتدريب (السعودية)*، "الرياضية" vol. 030, no. 059, 2014.
- [25] ا. م. ب. منشط، "تقرير عن ندوة علمية بعنوان: دور الإعلام الرياضي في الحد من التعصب والعنف في" *مجلة البحوث الامنية (السعودية)*، "الملاعب" vol. 023, no. 058, 2014.
- [26] *مجلة جيل*، "ن. عزوز، "دور الإعلام الرياضي المكتوب في التقليل من ظاهرة العنف في ملاعب كرة القدم" *العلوم الإنسانية والاجتماعية - مركز جيل البحث العلمي - الجزائر* vol. 000, no. 022, 2016.
- [27] "ج. ع. م. داود، "دور الصحافة الرياضية في الحد من ظاهرة التعصب الرياضي لدى الشباب الجامعي" *مجلة الحكمة للدراسات الإعلامية والاتصالية - مؤسسة كنوز الحكمة للنشر والتوزيع - الجزائر* vol. 001, no. 029, 2015.
- [28] ج. ع. م. داود، "دور الصحافة الرياضية الإفريقية في الحد من ظاهرة التعصب الرياضي لدى الشباب" *مركز البحوث والدراسات الإفريقية-جامعة إفريقيا العالمية-دراسات إفريقية (السودان)*، "الجامعي" vol. 000, no. 053, pp. 69-119, 2015.
- [29] *ندوة أمن الملاعب الرياضية - السعودية*، "ع. ا. عادل، "دور وسائل الإعلام في أمن الملاعب الرياضية" vol. 000, no. 000, 2000.
- [30] *مجلة*، "أ. أ. حارز، "إيجاد مستويات معيارية للتعصب الرياضي لمشجعي كرة القدم and أ. ن. ك. جواد" vol. 10, no. 59, pp. 158-172, 2014.
- [31] *المؤتمر*، "ح. خ. م. حسن، "بناء مقياس للعوامل النفسية المؤثرة على ممارسة و انسحاب النشء للرياضة" *العلمي الدولي الثالث عشر (التربية البدنية والرياضة - تحديات الألفية الثالثة) - مصر* vol. 003, no. 000, 2010.

- [32] *المجلة العلمية للبحوث*, "إ.إ. عبدالعزیز, "بناء مقياس القيم الخلقية والاجتماعية للاعبى كرة القدم *والدراسات في التربية الرياضية (كلية التربية الرياضية ب بورسعيد) - مصر*, vol. 000, no. 002, 2001.
- [33] خ.إ. عبدربه, "المؤشرات السلوكية الدالة على التعصب الرياضي لدى بعض جمهور المشاهدين : بناء *المجلة العلمية للتربية البدنية والرياضة - مصر*", مقياس دراسة تحليلية, vol. 000, no. 054, 2008.
- [34] F. Amiri, S. Scerri and M. Khodashahi, "Lexicon-based sentiment analysis for Persian Text," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2015.
- [35] H. K. Aldayel and A. M. Azmi, "Arabic tweets sentiment analysis--a hybrid scheme," *Journal of Information Science*, vol. 42, no. 6, pp. 782--797, 2016.
- [36] F. H. Mahyoub, M. A. Siddiqui and M. Y. Dahab, "Building an Arabic sentiment lexicon using semi-supervised learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 4, pp. 417--424, 2014.
- [37] S. Mohammad, M. Salameh and S. Kiritchenko, "Sentiment Lexicons for Arabic Social Media," in *LREC*, 2016.
- [38] S. R. El-Beltagy and A. Ali, "Open issues in the sentiment analysis of Arabic social media: A case study," in *2013 9th International Conference on Innovations in Information Technology (IIT)*, 2013.
- [39] H. ElSahar and S. R. El-Beltagy, "A fully automated approach for arabic slang lexicon extraction from microblogs," in *International conference on intelligent text processing and computational linguistics*, 2014.
- [40] S. El-Beltagy and A. Ali, "unWeighted Opinion Mining Lexicon (Egyptian Arabic)," [Online]. Available: <http://bit.ly/MGtMqU>. [Accessed 4 April 2019].
- [41] S. El-Beltagy and A. Ali, "unWeighted Opinion Mining Lexicon (Egyptian Arabic)," 2013.
- [42] H. ElSahar and S. R. El-Beltagy, "Building large arabic multi-domain resources for sentiment analysis," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2015.
- [43] S. R. El-Beltagy, "NileULex: A Phrase and Word Level Sentiment Lexicon for Egyptian and Modern Standard Arabic.," in *LREC*, 2016.
- [44] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word--emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436--465, 2013.
- [45] S. R. El-Beltagy, "WeightedNileULex: A Scored Arabic Sentiment Lexicon for Improved Sentiment Analysis," in *Language Processing, Pattern Recognition and Intelligent Systems. Special Issue on Computational Linguistics, Speech & Image Processing for Arabic Language*. World Scientific Publishing Co, 2017.
- [46] M. Youssef and S. R. El-Beltagy, "MoArLex: An Arabic Sentiment Lexicon Built Through Automatic Lexicon Expansion," *Procedia computer science*, vol. 142, pp. 94-103, 2018.

- [47] T. Wilson, J. Wiebe and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.
- [48] "Translation," Google, [Online]. Available: <https://translate.google.com/>. [Accessed 4 April 2019].
- [49] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *LREC*, Citeseer, 2006, pp. 417--422.
- [50] E. Refaee and V. Rieser, "An arabic twitter corpus for subjectivity and sentiment analysis," in *LREC*, 2014.
- [51] F. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," *arXiv preprint arXiv:1103.2903*, 2011.
- [52] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.
- [53] S. Mohammad and P. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 2010.
- [54] S. M. Mohammad, S. Kiritchenko and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242*, 2013.
- [55] S. Kiritchenko, X. Zhu and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723--762, 2014.
- [56] M. Mataoui, O. Zelmati and M. Boumechache, "A proposed lexicon-based sentiment analysis approach for the vernacular Algerian Arabic," *Res. Comput. Sci*, vol. 110, pp. 55--70, 2016.
- [57] N. Al-Twairesh, H. Al-Khalifa and A. AlSalman, "Arasenti: Large-scale twitter-specific arabic sentiment lexicons," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.
- [58] A. Pasha, M. Al-Badrashiny, M. T. Mohamed, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow and R. Roth, "Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic," in *LREC*, 2014.
- [59] S. M. Mohammad, M. Salameh and S. Kiritchenko, "How translation alters sentiment," *Journal of Artificial Intelligence Research*, vol. 55, pp. 95--130, 2016.
- [60] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan and C. Callison-Burch, "Machine translation of Arabic dialects," in *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2012.
- [61] H. A. Alshahrani and A. C. Fong, "Arabic Domain-Oriented Sentiment Lexicon Construction Using Latent Dirichlet Allocation," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 2018.

- [62] B. Ihnaini and M. Mahmuddin, "An expandable and up-to-date lexicon for sentiment analysis of Arabic tweets," *Compusoft*, vol. 7, p. Compusoft, 2018.
- [63] S. El-Beltagy, "Nile University's Arabic sentiment Lexicon," [Online]. Available: <https://github.com/NileTMRG/NileULex>. [Accessed 4 April 2019].
- [64] A. Assiri, A. Emam and H. Al-Dossari, "Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis," *Journal of Information Science*, vol. 44, no. SAGE Publications Sage UK: London, England, pp. 184--202, 2018.
- [65] T. Al-Moslmi, M. Al-Bared, N. Omar, S. Abdullah and A. al-Shabi, "Multi-domain Arabic Sentiment Corpus (MASC)," [Online]. Available: <https://github.com/almoslmi/masc>. [Accessed 4 April 2019].
- [66] M. Saif, "Arabic Sentiment Analysis and Cross-lingual Sentiment Resources," [Online]. Available: <http://www.saifmohammad.com/WebPages/ArabicSA.html>; <http://www.purl.org/net/ArabicSA>. [Accessed 4 April 2019].
- [67] N. AlTwaresh, H. AlKhalifa and A. AlSalman, "AraSenti Lexicon," [Online]. Available: <https://github.com/nora-twaresh/AraSenti>. [Accessed 4 April 2019].
- [68] H. ElSahar and S. El-Beltagy, "Large Arabic Resources For Sentiment Analysis," [Online]. Available: <https://github.com/hadyelsahar/large-arabic-sentiment-analysis-resouces>. [Accessed 4 April 2019].
- [69] G. Badaro, R. Baly, H. Hajj, N. Habash and W. El-Hajj, "A large scale Arabic sentiment lexicon for Arabic opinion mining," in *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, 2014.
- [70] H. El Hajj, K. Shaaban, W. El Hajj, N. Habash, S. Elbassuoni and K. McKeown, "Opinion mining for Arabic (OMA)," [Online]. Available: <http://oma-project.azurewebsites.net/>. [Accessed 4 April 2019].
- [71] M. Abdul-Mageed and M. Diab, "Toward building a large-scale Arabic sentiment lexicon," in *Proceedings of the 6th international global WordNet conference*, 2012.
- [72] "Colloquial to Arabic Converter," Microsoft, [Online]. Available: <https://www.microsoft.com/en-us/research/project/colloquial-to-arabic-converter/>. [Accessed 4 April 2019].
- [73] H. E. Mahgouba and Y. Shaabanb, "A translator for arabic dialects to modern standard Arabic," in *The International Workshop on Computers and Information Sciences (WCIS)*, At Tabuk, Kingdom of Saudi Arabia, 2015.
- [74] A. Mountassir, H. Benbrahim and I. Berrada, "An empirical study to address the problem of unbalanced data sets in sentiment classification," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp 3298--3303, Seoul, Korea (South), 2012.
- [75] A. Elouardighi, M. Maghfour and H. Hammia, "Collecting and Processing Arabic Facebook Comments for Sentiment Analysis," in *International Conference on Model and Data Engineering*, Springer, 2017, pp. 262--274.
- [76] H. Abdellaoui and M. Zrigui, "Using Tweets and Emojis to Build TEAD: an Arabic Dataset for Sentiment Analysis," *Computacion y Sistemas*, vol. 22, 2018.

- [77] "Emojitracker for realtime emoji use on twitter," [Online]. Available: <http://emojitracker.com/>. [Accessed 4 April 2019].
- [78] N. Kralj , S. Jasmina , S. Borut and M. Igor, "Emoji Sentiment Ranking v1.0," [Online]. Available: http://kt.ijs.si/data/Emoji_sentiment_ranking/. [Accessed 4 April 2019].
- [79] M. Nabil, M. Aly and A. Atiya, "Astd: Arabic sentiment tweets dataset," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.
- [80] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman and Y. Al-Ohali, "Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets," in *Procedia Computer Science*, 2017.
- [81] N. A. Abdulla, N. A. Ahmed, M. A. Shehab and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," in *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, 2013.
- [82] E. Refaee, "Sentiment analysis for micro-blogging platforms in Arabic," in *International Conference on Social Computing and Social Media*, 2017.
- [83] M. Salameh, S. Mohammad and S. Kiritchenko, "Sentiment after translation: A case-study on arabic social media posts," in *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2015.
- [84] M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López and J. M. Perea-Ortega, "OCA: Opinion corpus for Arabic," *Journal of the Association for Information Science and Technology*, vol. 62, no. 10, pp. 2045--2054, 2011.
- [85] Z. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, O'Reilly Media, Inc., 2018.
- [86] I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh, "Feature extraction: foundations and applications," vol. 207, 2008.
- [87] G. Dong and H. Liu, *Feature engineering for machine learning and data analytics*, CRC Press, 2018.
- [88] L. M. Oberman and P. G. Enticott, *Neurotechnology and Brain Stimulation in Pediatric Psychiatric and Neurodevelopmental Disorders*, Academic Press, 2018.
- [89] "BusinessDictionary," [Online]. Available: <http://www.businessdictionary.com/definition/engineering.html>. [Accessed 4 April 2019].
- [90] "What is Engineering," [Online]. Available: <http://whatisengineering.com/>. [Accessed 4 April 2019].
- [91] D. Sarkar, R. Bali and T. Sharma, "Feature Engineering and Selection," in *Practical Machine Learning with Python*, 2018.
- [92] N. A. Zaidi, *Feature Engineering in Machine Learning*, 2015.
- [93] S. Chakraborty, S. Paul, R. Sarkar and M. Nasipuri, "Feature Map Reduction in CNN for Handwritten Digit Recognition," in *Recent Developments in Machine Learning and Data Analytics*, 2019.

- [94] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil and D. S. Turaga, "Learning Feature Engineering for Classification," in *IJCAI*, 2017.
- [95] N. Al-Twairesh, H. Al-Khalifa, A. Alsalman and Y. Al-Ohali, "Sentiment analysis of arabic tweets: Feature engineering and a hybrid approach," *arXiv preprint arXiv:1805.08533*, 2018.
- [96] M. Mourad and K. Darwish, "Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs," in *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 2013.
- [97] A. Abbasi, H. Chen and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, p. 12, 2008.
- [98] S. Al-Azani and E.-S. El-Alfy, "Emojis-Based Sentiment Classification of Arabic Microblogs Using Deep Recurrent Neural Networks," in *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, 2018.
- [99] M. Hammad and M. Al-awadi, "Sentiment analysis for Arabic reviews in social networks using machine learning," in *Information Technology: New Generations*, Springer, 2016, pp. 131--139.
- [100] S. Abuelenin, S. Elmougy and E. Naguib, "Twitter Sentiment Analysis for Arabic Tweets," in *International Conference on Advanced Intelligent Systems and Informatics*, Springer, 2017, pp. 467--476.
- [101] A. Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in *International Conference on Collaboration Technologies and Systems (CTS)*, IEEE, 2012, pp. 546--550.
- [102] A. M. Shoukry, "Arabic Sentence-level sentiment analysis- Thesis," American University in Cairo. Dept. of Computer Science and Engineering, Cairo, 2013.
- [103] A. M. AlAsmar, "Feature Based Approach in Arabic Opinion Mining Using Ontology," The Islamic University, Gaza, 2016.
- [104] L. Abd-Elhamid, D. Elzanfaly, Eldin and A. Sharaf, "Feature-based sentiment analysis in online Arabic reviews," in *11th International Conference on Computer Engineering & Systems (ICCES)*, IEEE, 2016, pp. 260--265.
- [105] S. Al-Azani and E.-S. M. El-Alfy, "Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text," in *Procedia Computer Science*, 2017.
- [106] S. Al-Azani and E.-S. M. El-Alfy, "Hybrid deep learning for sentiment polarity determination of arabic microblogs," in *International Conference on Neural Information Processing*, 2017.
- [107] S. Al-Azani and E.-S. M. El-Alfy, "Combining emojis with Arabic textual features for sentiment classification," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018.
- [108] "Twitter Developers," [Online]. Available: <https://developer.twitter.com/>. [Accessed 4 April 2019].

- [109] "QCRI Arabic Language Technologies Tools & Demos "FARASA"," [Online]. Available: <http://qatsdemo.cloudapp.net/farasa/>. [Accessed 4 April 2019].
- [110] R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald and D. Scuse, WEKA Manual for Version 3-9-2, Hamilton, New Zealand: University of Waikato, 2017.
- [111] م. ح. علاوي، سيكولوجية العدوان والعنف في الرياضة، القاهرة: مركز الكتاب للنشر، 2004.
- [112] "Twitter," [Online]. Available: <https://twitter.com/>. [Accessed 4 April 2019].
- [113] "Twitter reaches half a billion accounts More than 140 millions in the U.S.," SemioCast the Social Media Intelligence Company , [Online]. Available: https://semioCast.com/en/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US. [Accessed 4 April 2019].
- [114] "Twitter APIs Pricing," [Online]. Available: <https://developer.twitter.com/en/pricing.html>. [Accessed 4 April 2019].
- [115] S. Liu and K. Oliver, "A Java HTTP client for consuming Twitter's standard Streaming API," Twitter4j, [Online]. Available: <https://github.com/twitter/hbc>. [Accessed 4 April 2019].
- [116] "MongoDB NoSQL Database," [Online]. Available: <https://www.mongodb.com/>. [Accessed 4 April 2019].
- [117] "Studio 3T is the professional MongoDB GUI and IDE," [Online]. Available: <https://studio3t.com/>. [Accessed 4 April 2019].
- [118] "Standard streaming API request parameters," Twitter, [Online]. Available: <https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters.html>. [Accessed 4 April 2019].
- [119] C. Stanford, "The Stanford CoreNLP Natural Language Processing Toolkit," Stanford, 3 4 2017. [Online]. Available: <https://github.com/stanfordnlp/CoreNLP/blob/master/src/edu/stanford/nlp/trees/international/arabic/ArabicUtils.java>. [Accessed 22 8 2019].
- [120] "MADAMIRA Tool for Morphological Analysis and Disambiguation of Arabic," [Online]. Available: <https://camel.abudhabi.nyu.edu/madamira/>. [Accessed 4 April 2019].
- [121] "Assem's Arabic Light Stemmer (BETA)," [Online]. Available: <https://www.arabicstemmer.com/>. [Accessed 4 April 2019].
- [122] "Stemming and Lemmatization with Python NLTK," [Online]. Available: <https://text-processing.com/demo/stem/>. [Accessed 4 April 2019].
- [123] M. Boudchiche, A. Mazroui, M. Bebah, A. Lakhouaja and A. Boudlal, "AlKhalil2," [Online]. Available: <http://toolkit.oujda-nlp-team.net/AlKhalil2/>. [Accessed 4 April 2019].
- [124] M. Boudchiche , A. Mazroui , M. Bebah , A. Lakhouaja and A. Boudlal, "AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer," vol. 29, no. 2, pp. 141 - 146, 2017.

- [125] "Advance Search," Twitter, [Online]. Available: <https://twitter.com/search-advanced?lang=en>. [Accessed 4 April 2019].
- [126] "Saudi Arabian Football Federation," SAFF, [Online]. Available: <http://www.saff.com.sa/ar-sa/rules-regulations>. [Accessed 4 April 2019].
- [127] "Almaany Dictionary," [Online]. Available: <https://www.almaany.com>. [Accessed 4 April 2019].
- [128] Available: [. \[متصل\]. \https://www.almaany.com/ar/thes/ar-ar
[تاريخ الوصول 4 أبريل 2019].
- [129] Y. Niwa and Y. Nitta, "Co-occurrence vectors from corpora vs. distance vectors from dictionaries," in *Proceedings of the 15th conference on Computational linguistics*, 1994.
- [130] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, 1990.
- [131] A. Mohammed, "Proposed web-application for annotation," Alqmase, 2019. [Online]. Available: <https://youtu.be/9CQaVTRA6D8>. [Accessed 1 4 2019].
- [132] B. Chandra and M. Gupta, "An efficient statistical feature selection approach for classification of gene expression data," *Journal of Biomedical Informatics*, vol. 44, no. 4, pp. 529-535, 2011.
- [133] A. Mountassir, H. Benbrahim and I. Berrada, "An empirical study to address the problem of unbalanced data sets in sentiment classification," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp 3298--3303, IEEE, 2012.
- [134] T. Al-Moslmi, M. Albared, A. Al-Shabi, N. Omar and S. Abdullah, "Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis," *Journal of Information Science - SAGE Publications Sage UK: London, England*, vol. 0, no. 0, 2017.
- [135] R.-E. Fan, K.-W. Chang, C.-j. Hsieh, X.-R. Wang and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research*, vol. 9, pp. 1871--1874, 2008.
- [136] H. Mohammad and S. MN, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 2, 2015.
- [137] M. Abdul-Mageed and M. Diab, "Subjectivity and sentiment annotation of modern standard arabic newswire," in *Proceedings of the 5th linguistic annotation workshop*, 2011.
- [138] M. El-Masri, N. Altrabsheh, H. Mansour and A. Ramsay, "A web-based tool for Arabic sentiment analysis," *Procedia Computer Science- Elsevier*, no. 117, pp. 38--45, 2017.
- [139] M. Nabil, M. Aly and A. Atiya, "LABR: A Large Scale Arabic Sentiment Analysis Benchmark," *arXiv preprint arXiv:1411.6718*, 2014.

Vitae

Name :Mohammed Mugahed Yahya Alqmase |

Nationality :Yemeni |

Date of Birth :1/7/1984|

Email :alqumasi@gmail.com|

Address :Allawqia street, Sana'a, Yemen|

Academic Background :Received an M.S in Computer Science from King Fahd University of petroleum & Minerals, Saudi Arabia in 2019. He holds a B.S. Degree in Information Technology (IT) from King Abdul-Aziz University which was conferred in 2013. He worked as a content management system analyst for Hippo CMS in 2017. He also worked as an instructor in Sana'a Community College from 2013-2015. His research interests include sentiment analysis, natural language processing machine learning, algorithms and software engineering. .|