Arabic Sentiment Analysis using Machine Learning

تحليل المشاعر فى اللغة العربية باستخدام تعليم الآلة

by

SASI FUAD ATIYAH

Dissertation submitted in fulfilment

of the requirements for the degree of

MSc INFORMATICS

at

The British University in Dubai

September 2016

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

____

Signature of the student

COPYRIGHT AND INFORMATION TO USERS

Abstract

Sentiment Analysis is a rising field that is gaining popularity every day due to its importance in mining the public opinions, the immense amount of generated data every second over the Internet via social network, microblogs, blogs, forums, consumer websites and other presents a rich field of opinions that are ready to be populated, aggregated and summarized and based on that decision are made. The applications are wide from the classical problems like political campaigns, product reviews to more sophisticated usage in Human Machine Interaction where the detection of the human sentiment plays an important role in a successful machine interaction. In this research we investigated the problem of sentiment analysis in the Arabic language and focus on how to utilize the machine learning-based approach to its maximum by conducting several experiments on several multi-domain dataset and optimize the trained model using parameter optimization and using the findings to establish a predefined best parameter settings to be used on new datasets. The research showed that through parameter optimization, basic machine learning classifiers achieved higher results than other more complex hybrid approaches, in addition, the overall parameters settings were tested on two new datasets and provided very promising results indicating that performance weren't as a cause of overfitting. The research also explains the issues of testing such well-trained models on an unseen dataset from different sources in the same domain and how it can be solved. The work was concluded by the possible enhancements that can be applied to the work done and a new path for future work that promises a more generalized solution.

الخلاصة

يُعد تحليل المشاعر مجالًا صاعدًا يكتسب شعبية كل يوم بسبب أهميته في استخراج الآراء العامة والكم الهائل من البيانات التي يتم إنشاؤها كل ثانية عبر الإنترنت عبر شبكات التواصل الاجتماعي والمدونات والمنتديات والمواقع الإلكترونية للمستهلكين وغيره حيث يتم تجميع وتلخيص هذه الآراء ليتم آخذ قرارات. التطبيقات فى هذا المجال واسعة الأمثلة النموذجية تحديد الآراء فى الحملات السياسية ، ومراجعة نقد المنتجات إلى الاستخدام الأكثر تطوراً في التفاعل بين الإنسان والآلة حيث يلعب اكتشاف المشاعر الإنسانية دورًا مهمًا في تفاعل الآلة الناجح. في هذا البحث ، قمنا بدراسة مشكلة تحليل المشاعر في اللغة العربية والتركيز على كيفية استخدام النهج القائم على التعلم الآلي إلى أقصى حد من خلال إجراء العديد من التجارب على العديد من مجموعات البيانات متعددة المجالات وتحسين النموذج المدرب من خلال تحسين قيم المعاملات (المتغيرات) واستخدام النتائج لتأسيس وبناء أفضل الإعدادات للنماذج التى تم تدريبها لاستخدامها في مجموعات البيانات الجديدة. أظهر البحث أنه من خلال تحسين قيم المعاملات (المتغيرات) ، حققت نماذج التصنيف الأساسية للتعلم الآلي نتائج أعلى من الأساليب الهجينة الأكثر تعقيدًا ، بالإضافة إلى ذلك ، تم اختبار إعدادات المعاملات الشاملة في مجموعتي بيانات جديدتين ووفرت نتائج واعدة جدًا تشير إلى أن الأداء لم يكن سببًا overfitting . يشرح البحث أيضًا مشكلات اختبار هذه النماذج المدربة جيدًا على مجموعة بيانات غير مستخدمة سابقاً من مصادر مختلفة في نفس المجال وكيف يمكن حلها. تم الانتهاء من البحث مع شرح للتحسينات الممكنة التي يمكن تطبيقها على العمل المنجز ومسار جديد للعمل في المستقبل الذي يبشر بحل أكثر تعميما.

Dedication

To My Partner

To My Supporter

To My Motivator

…

To The One Who made this possible

…

My Beloved Wife

…

Maisoon

Acknowledgements

Foremost, I am thankful to Allah for blessing with the chance to complete my studies, and I hope I have done right by this blessing and will be able to contribute any possible new knowledge to the academic community.

My sincere thanks, gratitude goes to my supervisor, Prof Khaled Shaalan, for his tremendous support and encouragement throughout my studies, not just the work presented here, he has fuelled my passion to academic research and made sure always to direct me with his incredible experience and wisdom.

My thanks also go to Dr. Sherief Abdallah as his tutored modules and ability to simplify the complex world of machine learning made this work possible without any doubt.

Special thanks go to Miss Asmaa Mountassir and Mr Amine Bayoudhi, whom both thankfully supported my work by providing datasets to run this research experiments.

And finally, a thank you to all BUiD for such an amazing experience for the last three years, staff and students.

Table of Contents

List of Tables

List of Figures

List of abbreviations

ML	Machine Learning
SA	Sentiment Analysis
SC	Sentiment Classification
OP	Opinion Mining
SVM	Support Vector Machine
NB	Naïve Bayes
MSA	Modern Standard Arabic
HMI	Human Machine Interaction
BoW	Bag of Words
SGD	Stochastic Gradient Descent
KNN	K-Nearest Neighbour
TO	Term Occurrence
BTO	Binary Term Occurrence
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
ANN	Artificial Neural Networks
NN	Neural Networks
DL	Deep Learning

Chapter 1

Introduction

Opinions play a major role in our lives, based on opinions we make our decisions; these point of views can be our own thoughts or beliefs towards a subject based on experience, ideas or simply a feeling. Sometimes we face difficulties formulating an opinion on a matter at hand, so we seek advice, a second opinion, from friends, relatives, or most probably we google it!

The need of opinions is evident in everyday life, which road to take to avoid traffic jam? What is the best smartphone in the market based on my needs? Whom shall I vote for? etc. Also, the need for opinions is not limited to individuals, but also manufactures designing a new product, a company launching a new website, a politician seeking public approval.

So, the question is, how many opinions can we get? Where can we get them? And how accurate are they? Currently now a day there is no shortage of opinions, everyone is expressing themselves on almost every topic that occurs, and such opinions can be found online in different formats, blogs, microblogs, forums, reviews, etc. Now whether those opinions are correct or not is impossible to know, but what those opinions imply? That's what count. Hence the field of Opinion Mining and Sentiment Analysis was born. Opinion Mining (OM) or what is also known as Sentiment Analysis (SA) is defined as follow:

"*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.*" (Liu 2012)

The main focus of OM or SA is to detect the polarity of an opinion, in other words, does the opinion hold a positive emotion towards an entity or a negative one, the most common levels of polarity are positive, negative and neutral, there are other more complex forms of polarity classification that look into the different types of emotions using the hour glass of emotions(Cambria et al. 2012). The increasing amount of interest in SA is also due to the huge popularity of social networks, and information growth caused what we know today as Big Data. The most recent published info graph of Data Never Sleeps 4.0(DOMO 2016) shows the staggering amount of data generated in different social networks every minute by the total population of internet users that reached 3.4 billion.

This chapter will provide an overview of SA, the motivation behind the research, aims and objectives, the research questions to be answered and the dissertation structure.

1.1 An overview about Sentiment Analysis

With the continuous flow of information generated every day on the web, it is next to impossible to interpret the opinions manually, for example, to monitor the public opinion of a political event on Twitter, we might be required to read tens of thousands or in some cases in highly populated countries millions of tweets to reach a conclusion summarizing the opinion.

The sentiment classification problem has been the focus of research for some time and has increased in popularity with the information boom and growing computing power that facilitated more resources to carry out such research. The public opinion is a matter of concern to someone always, whether it’s a new product released, a movie at the box office, a political campaign…etc. People generously provide their opinions either through forums, blogs, social networks.

This wasn’t the case before the internet age, each party interested in finding out people’s opinion; an opinion poll must be conducted, either face-to-face or over the phone, results are later analyzed. Now a day, opinion mining data collection occurs in real time over social networks, microblogging sites and other. Several companies provide such service in different ways, but in general, there is a clear need for systems that supports Arabic language, for example when taking a closer look at the one of the most famous and prestigious available NLP tools, the Stanford CoreNLP, we find that sentiment analysis is covered only in English, even though the NLP library provides support for other languages covering other NLP topics, as a matter of fact, SA is only covered in English, and none of the other supported languages by the library are covered. Table 1 provides more details of the CoreNLP version 3.6.0. and the different NLP support.

Annotator	**AR**	**ZH**	**EN**	**FR**	**DE**	**ES**
Tokenize/Segment	✓	✓	✓	✓		✓
Sentence Split	✓	✓	✓	✓	✓	✓
Part of Speech	✓	✓	✓	✓	✓	✓
Lemma			✓			
Named Entities		✓	✓		✓	✓
Constituency Parsing	✓	✓	✓	✓	✓	✓
Dependency Parsing		✓	✓	✓	✓	
Sentiment Analysis			✓			
Mention Detection		✓	✓			
Coreference		✓	✓			
Open IE			✓			

Table 1: Stanford CoreNLP V3.6.0 Supported Languages

(Stanford 2016)

1.2 Motivations

As briefed in the introduction by the size and growth of Information, and the lake of support for the Arabic language lies the motivation behind this research. The ever growing demand and highly profitable potential income of such applications that are capable of classifying sentiments make such research of high demand. While there are several published papers on the matter, yet room for improvement is still available, as yet several challenges still face researchers, from the lake of linguistic resources, domain dependent solutions, handling different dialects. Arabic sentiment corpus is hard to come by, and few are available free for research making it a challenging task to build models the can be generalized even in the same domain. The challenge of language dependent is extremely a problem in

Arabic culture, as users tend to mix between Arabic and Language, making this a requirement for an effective SA application, it is still possible to focus on a more formal linguistic form of Arabic (Modern Standard Arabic), but that will limit the system for formal reviews by critics, journals, etc. and not be able to be utilized on the free chaotic social networks.

The sentiment classification problem is also dependent on the type of context, is it a document or a sentence? Does it contain more than one opinion holder and more than one entity? The list of criteria on which to build a sentiment classification application is further discussed in the Issues and Challenges section.

Having a system that is capable of classifying either a sentence or a document and at the same time regardless of the domain provide a very flexible functionality and scope. While the goals might be very ambient and hard to achieve, but as technology advancements and the incredible amount of data growth each day, developing such systems will be a must to find an efficient and quick way to sniff through the data, and to reach such goals, the smallest advancements can make that a reality, hence our focus in this research will be basically an attempt to achieve a significant increase in the accuracy of Arabic Sentiment Analysis, that does not mean to be a major increase, but more of sustained increase over multiple datasets trained and untrained.

1.3 Aim of Research

The literature shows two main approaches to solving the sentiment analysis problem, either by using a lexicon-based approach, or using a Machine Learning (ML) based approach. Off course, other variation had been researched using a hybrid of both approaches, but they remain at the core either a lexical based or ML base. New research has been published using Deep Learning (DL), an unsupervised method that is based on Neural Networks but has a more complex network structure. This approach was not possible in early 2000 due to several reasons such as the required computational power to train the model and no efficient learning algorithms, both of which have been solved in the recent years by the introduction of more efficient and faster algorithms, and more powerful computation hardware with the use Graphical Processing Units (GPU) to do the extensive matrixes calculations.

The field of SA has seen several attempts to solve the problem using DL with increased levels of accuracy(Irsoy & Cardie 2014)(Liu et al. 2015), but there has been only one research to our knowledge that investigated DL in Arabic language(Sallab et al. 2015).

The aim of this research is to focus on the Machine Learning based approach and try to achieve an increase in the classification accuracy on more than one dataset in the same domain.

1.4 Research Questions

The dissertation attempts to answer the following research question:

- RQ1: Is it possible through parameter optimization and a basic classifier achieve higher levels of performance compared to other more rich and hybrid approaches?
- RQ2: Can we establish a predefined parameter configuration based on parameter optimization for both feature vector generation and classifier configuration to be used on other dataset and achieve good results?
- RQ3: Can a classification model built using a domain specific sentiment corpus achieve comparable results on a blind data set in the same domain?

1.5 Methodology

Different feature vector generation methods were used to study the impact on the accuracy and whether it has a relevant statistical impact or not, using different term weighting approaches (Term Occurrence, Binary term Occurrences, Term Frequency, TF-IDF), Light Stemming, Stemming, bi-grams, and tri-grams.

The approach is to build a machine learning based classifier (SVM, NB) and increase its efficiency using parameter optimization.

This will be done for two datasets in the same domain (movie reviews), the accuracy of both models will be compared to see how much does the dataset affect the same classification model. (the results showed that using SVM with the same settings on two different corpora generated different accuracy levels indicating that the quality or content of the corpus will significantly impact the classifier performance)

The best-trained model (with the highest accuracy) from both experiments will be tested on an unseen dataset (the other dataset) and observe the accuracy level, based on this results RQ1 will be answered.

1.6 Dissertation Structure

The dissertation is organized as follow; chapter two will provide the literature review of sentiment analysis with a focus on research on the Arabic language. Chapter three explains the research methodology and experiment design, chapter four analysis of the results and findings; chapter five discuss possible enhancements of the research; chapter six provides the conclusion and future work.

Chapter 2

Literature Review

The literature review presents the different concepts behind the topic of sentiment analysis and the work done in this field with focus on research published around the use of Arabic Language. In addition to the different corpora used in the dissertation.

2.1 Basic Concepts

Before proceeding with the literature review, basic concepts will be briefly explained in order to provide an understanding of the topic. Sentiment Analysis is a natural language processing problem, and it shares many aspects with other NLP topics.

2.1.1 Bag of Words

Bag of Words (BoW) is a representation model of documents, where words are listed in no order but maintaining a count of each. It is used in document classification and the count of words or what is referred to also as the frequency of terms, represents a feature of the document that can be used as a training input when building a classifier. The generated BoW is used to create a documents term matrix, several approaches are used to represent the term frequency, a simple term occurrence count, a binary term occurrence where the count of words is ignored, a term frequency and mostly commonly used the term frequency inverse document frequency TF-IDF, where it addresses the problem of very popular words that don’t affect the context.

2.1.2 n-Grams

As explained in the BoW, the order of words is ignored and in a field of NLP the order holds significance, hence another representation model that retains the order was used, n-grams, where n words sequence frequency was captured, most used model are 2-gram(bi-grams) and 3-gram(tri-grams). In this way, the BoW can be thought of as an n-gram with n=1.

2.1.3 Tokenization

Is probably one of the very early steps before processing text, the process splits the document or sentence into a list of words called tokens based on separators, the separator in the simplest form is a white space, but it is more than that where we might need to split where based on punctuations and other special characters in order to allow for other NLP processes to handle the ward for further processing.

2.1.4 Part of Speech

PoS is the process where each token is identified based on its syntactic, if it is a noun, verb, adjective, etc. Such identification gives a breakdown of the document or sentence structure.

2.1.5 Stemming & Lemmatization

Both processes try to achieve the goal of reducing the word to its origin; the difference is stemming performs the process in a way that simply removes any extra affixes with the hope you are left with the word root, on the other hand, lemmatization does the process with the use of morphological analysis.

2.1.6 Stop Words

Stop words are basically common words that have no impact if removed on the meaning of the document, removing the stop words will help to reduce the size of the BoW and speed the calculation process.

2.1.7 Sentiment Lexicon

A sentiment lexicon is a dataset of sentiment words that hold a certain polarity, most commonly a list of positive and negative words, or more detailed form containing weak negative, strong negative, weak positive and strong positive words. Other variation more advanced lexicon were introduced like the work AreSenl(Badaro et al. 2014) where the lexicon contains for each word in the lexicon a positive score, a negative score. The sentiment lexicon is used in lexicon based SA and hybrid approaches where an overall score is calculated based on the availability of these sentiment words in the document or sentence.

2.1.8 Sentiment Corpus

A sentiment corpus is a dataset containing documents, sentence, tweets that have been previously assigned a polarity to be used as a training set for training classifiers, several corpora were used in this research and will be introduced later.

2.2 Sentiment Analysis and Opinion Mining

Understanding the opinion mining problem requires an understanding in general for what is an opinion? The Oxford dictionary defines an opinion as

"*A view or judgment formed about something, not necessarily based on fact or knowledge*," or it could be

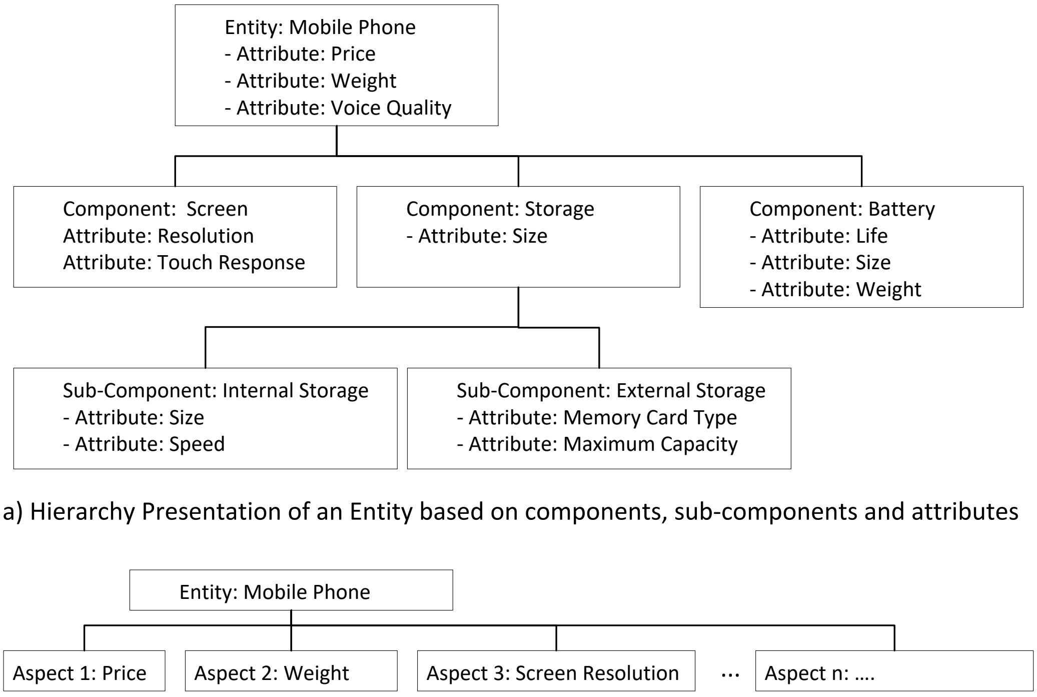
"*A statement of advice by an expert on a professional matter.*"

In our case of opinion mining and sentiment analysis, the former definition is related more closely to our problem. Having this broad definition requires a further analysis of what an opinion sentence or document can contain. In (Kim et al. 2004) defined an opinion as a quadruple [topic, holder, claim, sentiment], where the holder is the one holding a view or judgment [claim] about a [topic], and it is expressed via [sentiment] word, an adjective such as good or bad.

In a more recent study by (Liu & Zhang 2012), a further detailed definition of an opinion was stated to be as a quintuple [entity, aspect, orientation, holder, time]. This definition holds at a glance more obscure terms than the first definition which require further explanation; the authors used a product review example to derive their definition. The product review goes as follow:

“(1) I bought an iPhone a few days ago. (2) It was such a nice phone. (3) The touch screen was really cool. (4) The voice quality was clear too. (5) However, my mother was mad with me as I did not tell her before I bought it. (6) She also thought the phone was too expensive, and wanted me to return it to the shop . . .”(Liu 2010)

The review shows several opinions in sentences 2,3,4,5 and 6 that differ from positive to negative opinions, some regarding the phone, features of the phone or components of the phone, and in one case sentence 5 it was about the review himself, all these are considered to be target objects or defined as entities. An entity can be represented in a hierarchy structure, with the root node representing the entity itself, and all child nodes representing components or sub-components, each with any set of attributes as shown in figure 1(a). For the purposes of opinion mining and natural language processing, in general, the representation is simplified (flattened) in a way that all components, sub-components, and attributes/features are represented in one layer and named Aspects as shown in figure 1(b).



a) Hierarchy Presentation of an Entity based on components, sub-components and attributes

b) Simplified Presentation of an Entity based on Aspects

Figure 1: Entity Representation

Opinions can be of two types, either regular opinions or comparative opinions, the first is the one as we saw in sentences 2 and 3 for example, comparative opinions are based on a relation or difference between two entities. Regular opinions either could be positive, negative, neutral or a mix of positive and negative, this is referred to as the orientation or polarity of the opinion. Going back to the quintuple definition [entity, aspect, orientation, holder, time] the definition of an opinion is(Liu & Zhang 2012):

"*An opinion (or regular opinion) is a quintuple* $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$*, where* e_i *is the name of an entity,* a_{ij} *is an aspect of* e_i*,* oo_{ijkl} *is the orientation of the opinion about aspect* a_{ij} *of entity* e_i*,* h_k *is the opinion holder, and* t_l *is the time when the opinion is expressed by* h_k*.*"

This definition of an opinion in this manner presents the text in a structured form which in turn assist in the opinion mining process, but in order to get such information requires a complex understanding of the different aspect of natural language, which in turn presents a challenge building such system.

2.2.1 Subjectivity and Sentiment Analysis (SSA)

Several published papers address both tasks of Subjectivity and Sentiment Analysis; the subjectivity classification is the step where the system identifies the sentence being objective or subjective, the objective sentence usually holds a fact and carries no sentiment, on the other hand, the sentiment sentence might contain an opinion or not.

In (Abdul-Mageed et al. 2014), the authors propose SAMAR a system for SSA for Arabic social media genres. The system is a Machine Learning(ML) based using SVM light. Due to the lake of resources in Arabic social media, they created their own corpora compromised of four datasets, DARDASHA, TAGREED, TAHRIR, and MONTADA.

The morphological features used in the research are Word Forms, POS tagging, Unique, Polarity Lexicon(PL), Dialectal Arabic Feature and Genre Specific Features; the study showed that the use of POS had more effect on the subjectivity analysis while the lexemes worked better for sentiment analysis. They used a two-level SVM classifier, first was used for the subjectivity classification and the second for sentiment classification.

The research was followed by a second one addressing the issue of the lake of Arabic lexicon resources(Abdul-Mageed & Diab 2014), the authors presented SANA, a large scale multi-genre, multi-dialect lexicon for Arabic, that will be used to enhance the performance of SAMAR.

2.2.2 Sentiment Analysis Tasks

The core task or what is well known as sentiment analysis task is sentiments classification, where an opinionated document or sentence gets a polarity classification of positive or negative. Other tasks(Cambria et al. 2013) are agreement detection where it determines the level of agreement between two documents, the classification of multimedia based on the mood for the purposes of Human Machine Interaction(HMI), opinion holder extraction and subjectivity detection. This research focuses only on the sentiment classification tasks as other tasks require their own research and investigation.

2.2.3 Sentiment Classification

Sentiment classification is the process of classifying a document or sentence with a polarity level; this could be a simple positive, negative and neutral classification, or a more detailed level of emotions from sad, happy, fear, etc. The process goes through several steps based on the approach but in general, they follow a common method as shown in figure 2.

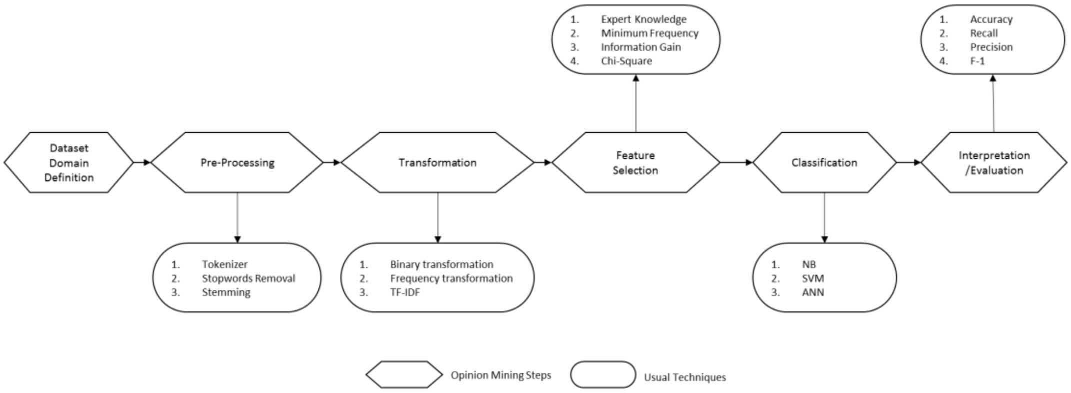


Figure 2: Steps & Techniques commonly used in SC approaches

(Moraes et al. 2013)

2.2.4 Sentiment Classification Levels

According to (Ganeshbhai & Shah 2015), sentiment classification can be performed on multiple levels, a document as a whole, a sentence, or further detailed level of a specific feature or aspect, the following is a brief description of each approach.

On the document level, opinion mining is performed on the whole review and it's either classified into positive or negative. For example, consider a movie review, based on the opinion words present in the review we classify the movie reviews as positive or negative. The main issue of this level is that a whole review is expressed on a single subject. Thus, it is not applicable to reviews in which single review expresses an opinion on multiple subjects.

In the case on the sentence level, the task of opinion mining is to categorize every sentence into a positive, negative, or neutral opinion. Sentences which contains no opinion or unrelated words are considered as neutral opinion. The sentence level opinion mining systems may contain subjectivity classification as the pre-processing step.

The last level is the Feature or Aspect Level, the document level, as well as the sentence level analysis, do not describe the exact liking of the people. Feature level opinion mining performs the finer-grained analysis. It is also referred as feature based or aspect-based opinion mining. Feature level analysis directly looks at the opinion itself instead of looking at language constructs like clauses, sentences or paragraphs. It builds on the fact that a user may express his opinion on specific feature or aspect of an entity but not the entity itself. Feature or aspect of an entity upon which opinion is expressed is referred as the target of an opinion.

In overall each of the sentiment classification levels focuses on a specific aspect of the sentiment analysis process, in table 2 we can see a matrix of each classification level and different tasks.

	Document Level	Sentence Level	Aspect Level
Subjectivity Detection		✓	
Opinion Holder Detection			✓
Polarity Detection	✓	✓	✓

Table 2: Classification level and Task Matrix

2.2.5 Sentiment Classification Approaches

In (Ganeshbhai & Shah 2015) surveyed the existing approaches used in sentiment classification and can be categorized broadly, as Machine Learning approach, dictionary-based approach, statistical approach and semantic approach. Table 3 shows a summary of algorithms or techniques used in research to solve the problem.

Machine Learning Approach	Lexicon-Based Approach
Supervised Learning - Linear Classification o Support Vector Machine o Neural Networks - Probabilistic Classifier o Naïve Bayes o Bayesian Network o Maximum Entropy - Decision Tree - Rule Based Classification - Deep Learning Unsupervised Learning - Deep Learning	Corpus-Based Approach – tries to find co-occurrence patterns of words to determine their sentiments - Semantic - Statistical Dictionary-Based Approach – uses synonyms, antonyms, and hierarchies in wordnet Case-Based Reasoning (CBR)

Table 3: Different Approaches and Techniques used in SA

2.3 Linguistic Resources

One of the main issues that have faced researchers in the field of Arabic SA is the difficulty of obtaining enough linguistic resource to be used in their experiments; the resources can be mainly divided into three sections, Arabic sentiment corpora, Arabic sentiment lexicon, and Arabic NLP tools. The majority of the research usually contains an initial phase of building a sentiment corpus or sentiment lexicon and sometimes both. The availability of such resources is not evident in the English language, so some attempts were made to use Machine Translation (MT) in order to benefit from these resources, but usually it comes with its issues as MT in itself is still an active research field. It is worth mentioning that currently exists some high volume, good quality resources but at a cost, making it difficult to obtain, but in the recent years a few corpora have been developed and made available for academic research, here we list what has been selected for this research and describe their characteristics briefly.

2.3.1 Sentiment Corpora

In the section the used corpus is reviewed, there exist other sentiment corpus but as explained previously they do come at a cost. The corpora are divided into two types, three of out of the five are document reviews the other two are tweets and sentence level. They also cover multiple domains from different sources. The used language is a mix of Modern Standard Arabic(MSA) and dialectical Arabic, the dialectical Arabic is more in use online and has no standard, words can be written in different spelling making it very hard for specific NLP processes like PoS.

2.3.1.1 OCA

The Opinion Corpus for Arabic (OCA)(Rushdi-Saleh et al. 2011) is a movie review document sentiment corpus collected from web pages and blogs, in language is in the form of dialectal Arabic. The corpus contains a balanced set of positive and negative reviews 500 in total. The corpus is available in the form of two labeled folders (positive and negative) each containing 250 text documents. The quality of the corpus suffers from the existing for English words that should require pre-processing, but experiments were conducted on them with the elimination of them. The corpus is probably one of the first that have been publicly and has been referenced in over 70 research papers. The corpus statistics can be seen in table 4.

	Negative	**Positive**
Total documents	250	250
Total tokens	94,556	121,392
Avg. tokens in each file	378	485
Total Sentences	4,881	3,137
Avg. sentences in each file	20	13

Table 4: Statistics of OCA Corpus

(Rushdi-Saleh et al. 2011)

2.3.1.2 ACOM

The Arabic Corpus for Opinion Mining built in (Mountassir et al. 2013) is a combination of three datasets (the reference describes only two, but through contacting the author a third dataset was obtained), the first dataset DS1 or DSMR is a movies review document sentiment corpus, the second DS2-DSSP a sport-specific dataset and the third DS3-DSPO focuses on political comments. The corpus is available in the same data format as OCA, where each class is a folder containing text files for each review/comment. In table 5 the overall breakdown of the corpus is listed, and it shows that it is an unbalanced dataset in general, in table 6 the statistics of the two classes, positive and negative are described by the number and percentage of the documents for each class and an average number of tokens.

Dataset	Positive	Negative	Neutral	Dialectal	Total Documents
DSMR	184	284	106	20	594
DSSP	486	517	391	98	1492
DSPO	149	462	383	88	1082

Table 5: Number of comments per category for each dataset of ACOM

Dataset	Positive		Negative		Total Documents
	Nb Doc	Avg tokens	Nb Doc	Avg tokens	
DSMR	184(39.4%)	60	284(60.6%)	63	368
DSSP	486(48.4%)	57	517(51.6%)	66	1003
DSPO	149(24.4%)	123	462(75.6%)	128	611

Table 6: Statistics of each dataset of ACOM

2.3.1.3 LABR

The Large-scale Arabic Book Review dataset(Nabil et al. 2014) is the third selected document sentiment corpus; the corpus contains 63,257 book reviews in both MSA and Dialectal form. The reviews have a rating from 1 to 5 starting from a strong negative to a strong positive and 3 as neutral. The corpus is also provided with a pre-list of training, testing and validation sampling for the goal of achieving a fair comparison when experimented by other researchers. Table 7 shows the corpus statistics as presented by the authors.

Number of reviews	63,257
Number of users	16,486
Avg. reviews per user	3.84
Median reviews per user	2
Number of books	2,131
Avg. reviews per book	29.68
Median reviews per book	6
Median tokens per review	33
Max tokens per review	3,736
Avg. tokens per review	65
Number of tokens	4,134,853
Number of sentences	342,199

Table 7: LABR Dataset Statistics

(Nabil et al. 2014)

2.3.1.4 ASTD

The Arabic Social Sentiment Twitter Dataset (Nabil et al. 2015) contains 10,006 tweets classified into an objective, positive, negative and neutral tweets. The corpus is in dialectal language form and not domain specific as the data collection of the tweets focused on the most active accounts and trending hashtags in Egypt, giving us an independent domain corpus. Table 8 shows the dataset statistics.

Number of Tweets	10,006
Median tokens per tweet	16
Max tokens per tweet	45
Avg. tokens per tweet	16
Number of tokens	160,206
Number of vocabularies	38,743

Table 8: ASTD Dataset Statistics

(Nabil et al. 2015)

2.3.1.5 OCA_GOLD and COPARD2_Gold

The authors in (A Bayoudhi et al. 2015) utilized two existing document sentiment corpus and generated a corpus at the sentence or discourse level, the use of the OCA corpus mentioned previously and the Corpus of Opinion Arabic Debates 2 (COPARD2). The corpora were broken down and ran through a multistep annotation process resulting in a gold standard dataset; the resulted dataset can be seen in table 9.

	OCA_GOLD	COPARD2_GOLD
Positive Segments	7,455	1,794
Negative Segments	4,931	1,110
Total	12,386	2,904

Table 9: Statistics of the Gold Standard versions of OCA and COPARD2

(A Bayoudhi et al. 2015)

2.3.1.6 Large Arabic Resources for Sentiment Analysis

This corpus is one of the latest that have been made publicly(ElSahar & El-Beltagy 2015), the corpus covers four domains, the domain of hotel reviews, restaurant reviews, movie reviews and product reviews. Table 10 shows a summary of the dataset statistics.

	HTL	RES#1	RES#2	MOV	PRO	All
# Reviews	15,579	8,664	2,646	1,524	14,279	42,692
# Unique Reviews	15,562	8,300	2,640	1,522	5,0092	33,116
# Users	13,407	1,639	1,726	416	7,465	24,653
# Items	8,100	3,178	1,476	933	5,906	19.593

Table 10: Summary of Dataset Statistics

(ElSahar & El-Beltagy 2015)

2.3.2 Sentiment Lexicons

As mentioned in a previous section a sentiment lexicon is a list of words that have been previously annotated with a polarity. The annotation takes different forms and purposes. In addition to the sentiment words, operators are included, operators are used to handling intensification, amplification, and negation, as each of them affects the polarity of the word in a different way. The approach is to give a score to each sentiment word, a +1 for positive word, a -1 for a negative word, the operators multiply these scores in different ways, the negation simply switches the polarity.

2.3.2.1 LAP

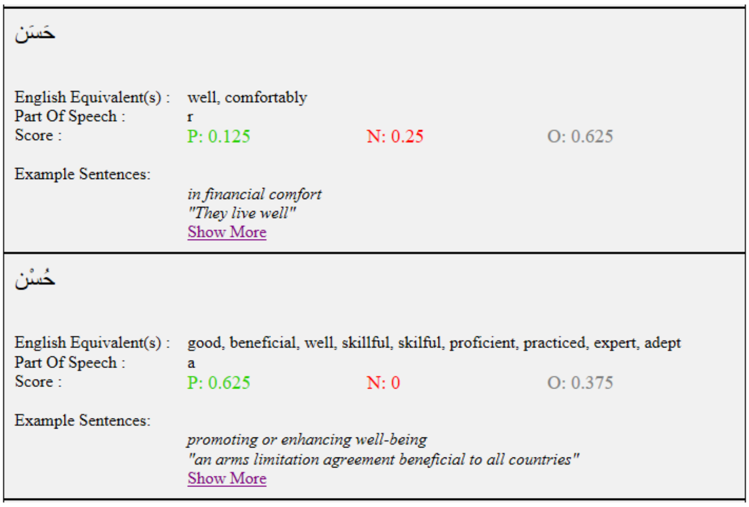
The authors in (A Bayoudhi et al. 2015) built in their work both a gold standard sentiment corpus mentioned in section 2.3.15 and a sentiment lexicon LAP(Lexicon of Arabic Polarized Words), the lexicon was built in a semi-automatic way by the use of several tools like ArabiWorNet, SentiStrenght and Linguistic Experts at the end. The lexicon was not built from scratch, but was based on the MPQA Arabic lexicon(Elarnaoty et al. 2012), the resulted lexicon contained 5,302 sentiment words divided into four classes, table 11 shows the breakdown of the lexicon.

Class	**Count**
Negative Strong	1,544
Negative Weak	1,719
Positive Strong	1,278
Positive Weak	761
Total	5,302

Table 11:LAP Breakdown

2.3.2.2 ArSenL

ArSenL (Arabic Sentiment Lexicon)(Badaro et al. 2014) is built based on previously existing lexicons and tools (ESWN, ArabicWordNet, and SAMA), mapping from Arabic to English in order to benefit from the existing resource, the authors goal was to build a rich, clear, large and publicly available sentiment lexicon. The authors have provided a web interface to browse the lexicon, in figure 3 we can see that the returned results for the word “حسن” has multiple matches (we list 2 but in fact it has 40) for different use of the word, we can see the positive score in the first example is lower than the negative, looking at the English example “They live well” might imply a slight dissatisfaction, but majority of the score goes to how objective the word is.



حَسَن

English Equivalent(s) : well, comfortably
Part Of Speech : r
Score : P: 0.125 N: 0.25 O: 0.625

Example Sentences:
in financial comfort
"They live well"
Show More

حُسْن

English Equivalent(s) : good, beneficial, well, skillful, skilful, proficient, practiced, expert, adept
Part Of Speech : a
Score : P: 0.625 N: 0 O: 0.375

Example Sentences:
promoting or enhancing well-being
"an arms limitation agreement beneficial to all countries"
Show More

Figure 3: Example using the ArSenl Web Interface

(OMA-Project 2016)

2.3.3 Tools

Several software tools exist that contain NLP functionality, the tool could be a dedicated NLP tool or as in most cases the NLP functionality is introduced as library or package, table 12 lists some of these tools.

Linguistic Tools
NLTK
GATE
OpenNLP
StanfordNLP
Opinion Finder
Ling Pipe
Review seer tool
Red Opal
Opinion Observer
Web Fountain
Weka
RapidMiner

Table 12: NLP Tools

2.3.4 Online Websites

Table 13 lists some of the available online sentiment analyzers; several other exist but with the limitation of not covering the Arabic Language.

Web Site	License	URL
30 dB	Free	http://www.30db.com/
AlchemyAPI	Commercial	http://www.alchemyapi.com/
BitextAPI	Commercial	https://www.bitext.com/
Etuma Oy	Commercial	http://www.etuma.com/home
HPE Haven OnDemand	Commercial	https://dev.havenondemand.com/apis/analyzesentiment
Semantria	Commercial	https://www.lexalytics.com
Sentiment140	Commercial	http://help.sentiment140.com/api
Stanford NLP	Academic	http://nlp.stanford.edu/sentiment/
Sentic API	Commercial	http://business.sentic.net/
Twinword	Commercial, Free	https://www.twinword.com/api/sentiment-analysis.php

Table 13: Online Sentiment Analyzers

2.4 Related Work

Here a review of related work will be examined, all of which focused on Arabic language and in specific those whom datasets that are available publicly or have been acquired through contacting the authors, and tested in this research. The focus has been limited to these research papers in order to establish a fair comparison and a valid way to explain the insight obtained from the results.

Rushdi-Saleh et al. (Rushdi-Saleh et al. 2011) has published one of the most referenced and used sentiment corpus, the OCA (Opinion Corpus for Arabic), it as a movies review sentiment corpus that contains 500 documents of balance positive and negative reviews (250 each), the authors experimented with a supervised sentiment analysis approach using Support Vector Machin (SVM) and Naïve Bayes, the pre-processing and feature extraction was in done in the standard way of tokenization, filtering stop word, stemming words, and addition of filtering words with length above 2 characters. The features extraction was done using n-grams(n=1,2,3). The reported results are showing a high accuracy level with more in favor for the use of SVM with trigrams and no stemming and word weighting of TF-IDF in a 10-fold cross validation; their results are shown in table 14.

Corpus	n-gram model	Precision	Recall	Accuracy
Pang	Unigram	0.8493	0.8390	0.8445
	Bigram	0.8583	0.8450	0.8515
	Trigram	0.8619	0.8450	0.8535
OCA	Unigram	0.8699	0.9480	0.9020
	Bigram	0.8738	0.9520	0.9060
	Trigram	0.8738	0.9520	0.9060

Table 14: Pang corpus 10-fold cross-validation results compared to OCA corpus best results.

(Rushdi-Saleh et al. 2011)

Mountassir, Benbrahim and Berrada(Mountassir et al. 2013) built the sentiment corpus ACOM (Arabic Corpus for Opinion Mining) in order to address the lack of Arabic resources in the area of sentiment analysis, their study investigated the use of machine learning based sentiment classifiers with the focus on Naïve Bayes, Support Vector Machine (SVM) and K-Nearest Neighbour. Their experiments were conducted on their data set ACOM and the OCA corpus; results showed that the use of light stemming is recommended with term occurrences for word weighting and a combination of unigrams and bigrams. They concluded that performance might be affected by document length, homogeneity, and source of documents, however, the size of the corpus has no impact.

Bayoudhi, Belguith and Ghorbel(Amine Bayoudhi et al. 2015) developed an ensemble-based classifier for document sentiment analysis; their approach focused on enhancing the used features vectors in training, instead of relying only on n-gram as in most literature they added the used of opinion and discourse features. Their systems performed the usual pre-processing steps of stemming and stop word removal in addition to a segmentation step, where the by the use of Stanford word segments, word normalization that handles the different possible spelling of a word. The opinion feature extraction was performed using the LAP sentiment lexicon, in where a list of sentiment words and their polarity have been identified. The classification algorithm used was a two-step process; first experiments were conducted to determine the best base classifier for each dataset, then based on that different ensemble techniques were used for the best-performed classifier. The used datasets in their experiments were the OCA corpus and the ACOM corpus. Reported results showed that they achieved an improved F-measure up to 4% by using the discourse feature, their results are an overall macro average of all tested datasets. The study showed the effect of using additional extracted features on the performance, but not tested their trained model on unseen datasets.

Atia and Shaalan (Atia & Shaalan 2015) investigated the possibility of increasing the accuracy of Arabic Sentiment Analysis using parameter optimization, the study focused on the OCA corpus and yielded noticeable increase in accuracy comparing to the results from (Rushdi-Saleh et al. 2011), the research also showed that certain kernel types when using SVM for classification did not yield acceptable results and that the ANOVA, polynomial, and dot kernel produced the best results.

The authors in(A Bayoudhi et al. 2015) tackled the sentiment analysis problem at the sentence level rather than on the document level, in order to do so they built both a sentiment lexicon LAP(Lexicon of Arabic Polarized Words) and a gold standard Arabic sentiment corpus (movie reviews and political debates), the corpus was based on an existing of two document level corpora, the OCA and COPARD2 (Corpus of Opinion Arabic Debates2), the authors developed a gold standard corpus after a process of segmentation and annotation. The sentiment analysis system built was a hybrid approach containing a lexicon detector for opinions and operators and supervised classifier. The achieved results can be seen in table 15.

Corpus	Accuracy	Precision	Recall	F-score
OCA_GOLD	70.48	67.91	87.18	76.35
COPARD2_GOLD	71.41	67.79	83.58	74.86

Table 15: Obtained results with the proposed methods

(A Bayoudhi et al. 2015)

Nabil, Aly and Atiya (Nabil et al. 2014) developed another sentiment corpus, the LABR(Large-scale Arabic Review) dataset consisting of over 63,000 book reviews, the corpus is valuable resource to the research committee, the dataset is provided with pre-split configuration for training, validation and testing making it more convenient in establishing a benchmark for SA systems performance testing. The authors tested with several classifiers used in the field of SA, like Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Support Vector Machine and others. The tests results were conducted only on the on a two-class case (positive and negative) although the dataset contains a rating prediction from one to five and can be categorized up to five classes (strongly negative, negative, neutral, positive and strongly positive), but that will make the classification process extremely hard and require a more complex approach for classifying a five-class case. The results reported are partially shown in table 16, with highest accuracy levels highlighted.

Classifier	TF-IDF	Balanced			Unbalanced		
		1g	1g+2g	1g+2g+3g	1g	1g+2g	1g+2g+3g
SVM	No	0.535/0.534	0.568/0.565	0.570/0.566	0.698/0.690	0.727/0.712	0.731/0.712
	Yes	0.566/0.564	**0.590/0.588**	**0.589/0.588**	**0.734/0.709**	**0.750/0.723**	**0.751/0.725**
Logistic Regression	No	0.570/0.568	0.586/0.583	0.590/0.585	0.728/0.707	0.743/0.717	0.737/0.703
	Yes	**0.587/0.583**	**0.590/0.588**	0.586/0.585	0.727/0.672	0.720/0.659	0.709/0.640

Table 16: Polarity Classification Experimental Results

(Nabil et al. 2014)

The authors (Nabil et al. 2015) developed an Arabic sentiment tweet dataset (ASTD) containing around 10,000 tweets that have been classified into four classes, objective, positive, negative and neutral. The experiments conducted were similar to their approach in (Nabil et al. 2014), but the results were best obtained using MNB and SVM instead of Logistic Regression and SVM, accuracy levels obtained were lower compared to the use of LAP, indicating the difficulties in classifying a more complex problem (4 classes) and using shorter documents (tweets).

Al Shboul, Al-Ayyoub and Jararweh (Al Shboul et al. 2015) investigated the multi-way sentiment classification using the LABR dataset; the accuracy results showed poor performance indicating the complexity of the problem and the need for an alternate approach. The work continued in (Al-ayyoub & Nuseir 2016) where they proposed a hierarchical classifier in the case of a multi-classification sentiment analysis, the approach works in 2 level hierarchal where the classifies the document as either positive, negative or neutral, then second level further classifies the document with a polarity of 1 or 2 if negative, 4 or 5 if positive. The 4 level hierarchal splits the decision on each polarity level. Table 17 shows the results achieved using this approach in comparison to a regular flat classifier, a very high increase in accuracy can be seen using the 4-level hierarchal specifically using the KNN and N.B.

Classifier	**2-level**	**4-level**
SVM	-1.2%	+3.7%
DT	+9.2%	+18.2%
NB	+4.6%	+28.1%
KNN	+19.7%	+49.7%

Table 17: Results

(Al-ayyoub & Nuseir 2016)

Elsahar and El-Beltagy (ElSahar & El-Beltagy 2015) built one of the most recent multi-domain sentiment corpora containing more than 33,000. The corpus covered the domain of hotel reviews, restaurant reviews, movie reviews and product reviews. They also investigated the best-used classifiers and used features. The best-performed classifier out of five tested. The Linear Support Vector Machine (Linear SVM) outperformed the Bernoulli Naïve Bayes, Logistic Regression, Stochastic Gradient Descent(SGD) and K-nearest neighbor; the results are shown in table 18 are the average accuracy.

Classifier	Accuracy	
	2 Classes	**3 Classes**
Linear SVM	0.824	0.599
Bernoulli NB	0.791	0.564
LREG	0.771	0.545
SGD	0.752	0.544
KNN	0.668	0.469

Table 18: Ranking of Classifiers by Average Accuracy

(ElSahar & El-Beltagy 2015)

In (Al-ayyoub et al. 2016) they used a popular lexicon-based SA tool, SentiStrength, the tool was tested to evaluate its effectiveness as it was originally designed for the English language, but it was not clear how effective is it to be used in the Arabic language. The evaluation conducted was tested on 11 corpora including LABR and ASTD. SentiStrength(Thelwall 2013) the Sentiment Strength detection tool, is a lexicon based SA tool built for the English language that also can handle other content than text like emoticons and exaggerated punctuations. The lexicon contains 2310 words with a positive polarity score rated from 1 to 5 and a similar negative polarity score from -1 to -5. The tool was designed in a way that can be customized for other languages and currently supports Arabic as well. Other features in the tool contain an idiom list, a spelling correction algorithm, a negation word list and an emoticon list with polarities. The results showed that SentiStrenght produced acceptable results similar to the ones achieved in English, table 19 list partial results of the authors focusing only on the dataset examined in this research (LABR and ASTD).

Dataset	Accuracy	Precision	Recall	F1	Negative Correct
D3: LABR	0.563	0.858	0.574	0.688	0.506
D4: ASTD	0.571	0.385	0.557	0.455	0.577

Table 19: The Results of All Datasets

(Al-ayyoub et al. 2016)

2.5 Conclusion

The field of Arabic SA has seen some active research in the past 5 years in attempts to catch up with research in the English language, major problems in this field is the poor availability of linguistics resources, specifically sentiment corpora as they are the main building block when training classifiers to build needed models. The process of building such corpora is expensive in terms of effort and time, the diversity of the domains and Arabic dialects makes it more complex. Researchers tend to create their own corpus on their research including a sentiment lexicon also if required before implanting their classification approach whether it is a machine learning based, lexicon or a hybrid. The issue visible of such research approach is that it’s not possible to compare results obtained from different studies since the experiment resources are different! So, experiments are conducted in a way a researcher establishes his own baseline or benchmark results implementing a standard approach to machine learning classification, then the proposed approach or enhancement is adding and results are compared. In all cases, an in an increase in accuracy is recorded, but in order to have a more tangible result, experiments should be conducted using the same resources in order to claim such increase in accuracy performance. Based on that the literature review focused on the research that has reused existing corpora in so it would be possible to compare their results. In this research, the contribution to Arabic Sentiment Analysis is the investigation of using a Machine Learning Approach and attempt to increase performance in accuracy using parameter optimization in a generalized way that can be reproducible on different corpora in different domains.

Chapter 3

Methodology and Experiment Design

This chapter will show the approach taken to investigate the effect of parameter optimization of machine learning algorithm used to solve Arabic Sentiment Analysis and whether those results match or close to other research that introduced additional functionality/steps into the classification process.

3.1 Methodology

The noticed thing with most approaches when solving Arabic Sentiment Analysis, they usually don’t 1) explore further performance enhancements through parameter optimization, 2) testing their models on blind datasets, and they fall into the error of 1)comparing obtained results to other results with experiment different settings(Rushdi-Saleh et al. 2011). 2) Averaging results obtained and indicated that there is an increase in performance. In some cases, the corpus is available with its actual author's sampling sets(Nabil et al. 2014) and (Nabil et al. 2015) giving other researchers the exact dataset characteristic and a way to have a fair comparison of results with others.

In this research, a solid, comprehensive benchmark of results will be generated with the focus of trying to achieve the best possible results through different feature vector generation techniques and model parameter optimization. Through the obtained results the comparison will then show how much accuracy performance has increased and compared it to other published research that used different or added steps to the process. Also produced models will be tested on completely unseen dataset from various sources giving an indicator on how far these models can be generalized and still achieve acceptable performance.

Before explaining the experiment design, it is important to have an understanding of the testing data and how it can assist in achieving our goal of answering the research questions. As explained in chapter two several sentiment corpora were selected based on the fact it has been used at least in two types of research in order to have a way to measure the actual increase in accuracy, out of the 13 datasets 6 were experimented at least twice. Table 20 is a matrix of the obtained sentiment corpora, and where they have been used, the star indicates that the mentioned research was the one that developed the corpus, at the bottom we have a total of how many times the corpus was used in experiments by different authors and different approaches. The highlighted datasets are the ones used in this research.

		1	2	3	4	5	6	7	8	9	10	11	12	13
		OCA	**ACOM-DSMR**	**ACOM-DSSP**	**ACOM-DSPO**	**LABR**	**ASTD**	**OCA_GOLD**	**COPGRD2_GOLD**	**HTL**	**MOV**	**PRO**	**RES#1**	**RES#2**
1	(Rushdi-Saleh et al. 2011)	✓ *												
2	(Mountassir et al. 2013)	✓	✓ *	✓ *	✓ *									
3	(Nabil et al. 2014)					✓ *								
4	(Nabil et al. 2015)						✓ *							
5	(A Bayoudhi et al. 2015)							✓ *	✓ *					
6	(ElSahar & El-Beltagy 2015)									✓ *	✓ *	✓ *	✓ *	✓ *
7	(Al-ayyoub et al. 2016)					✓	✓							
8	(Al Shboul et al. 2015)					✓								
9	(Al-ayyoub & Nuseir 2016)					✓								
10	(Amine Bayoudhi et al. 2015)	✓	✓	✓	✓									
11	(Atia & Shaalan 2015)	✓												
Total usage		4	2	2	2	4	2	1	1	1	1	1	1	1

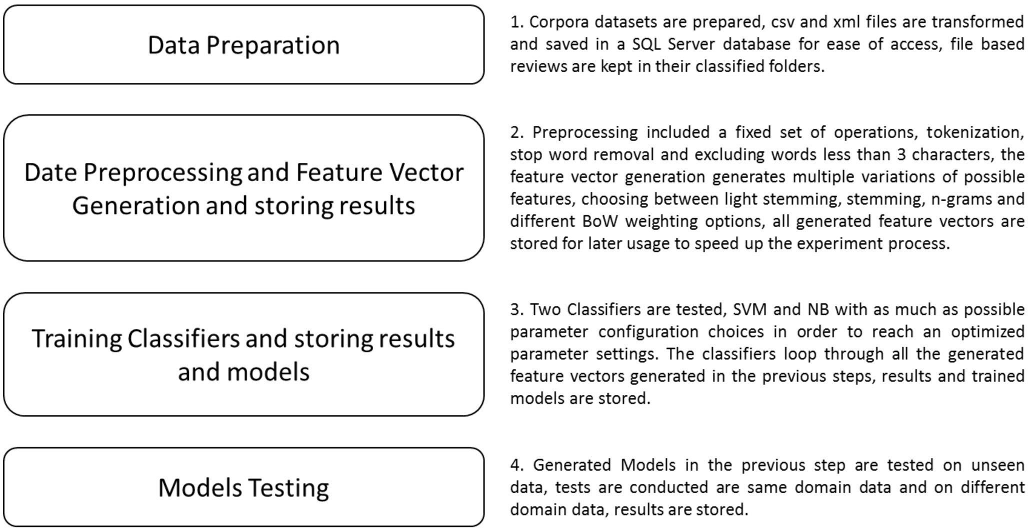
Table 20: Research Sentiment Corpora Usage

Table 21 shows the collected corpora category based on their domain, the purpose of this is to see what are the available options we have in testing our built models on an unseen dataset from a different source.

Domain	**Datasets**	**Count**
Movie Reviews	OCA,DSMR,MOV, OCA_GOLD	4
Politics	COPARD2_GOLD, DSPO	2
Restaurant Reviews	RES#1, RES#2	2
Books Reviews	LABR	1
Sports	DSSP	1
Hotel Reviews	HTL	1
Products Reviews	PRO	1
Tweets	ASTD	1
Total		13

Table 21: Corpora Domain

The overall experiment goes into four phases,1) Data Preparation Phase: here the collected corpora get prepared for processing, if the corpus is provided in a file base format where each file contains a review and is classified, no processing is required, for the corpus that is provided in the form of CSV files or XML files, the data was transformed and imported into SQL Server 2012 Developer edition in a tabular format, mainly with a two column structure one containing the review and the second containing the polarity/label/class/rating. 2) Date Preprocessing and feature vector generation phase: starts by retrieving the corpora data from either their files or database, the document processing steps include tokenization, stop word removal and filtering words that are less than 3 characters. The feature vector generation will produce multiple possible variations using different settings, by using light stemming, stemming, n-grams and different word weighting approaches, all generated results are saved into a repository for later use, this way the processing time is done once and when conducting training feature vectors are ready for use. 3)Training Classifiers Phase: in this phase the experiment is building models using the previously generated feature vectors, the classifiers chosen are SVM and NB, both classifiers are executed several times on each feature vector using different parameter settings, all generated results are stored in the repository in order to investigate the training performance and find the optimal feature vector selection and classifier parameter settings. 4) Models Testing Phase: at this point many researchers don't proceed on testing their generated models on unseen data, and in cases where it has been done was by splitting the original corpus into a training set, testing set and a validation set, the drawback of this approach is that even if the validation set is considered unseen, it is still originated from the same source, in order to have a true performance check the unseen data should come from a different source. The model testing will be done on corpora in the same domain and as an additional investigation on corpora in different domains giving an insight on is it possible to generalize such models. Figure 4 shows the different phases sequence with a brief description for each phase.


Data Preparation
Date Preprocessing and Feature Vector Generation and storing results
Training Classifiers and storing results and models
Models Testing

1. Corpora datasets are prepared, csv and xml files are transformed and saved in a SQL Server database for ease of access, file based reviews are kept in their classified folders.

2. Preprocessing included a fixed set of operations, tokenization, stop word removal and excluding words less than 3 characters, the feature vector generation generates multiple variations of possible features, choosing between light stemming, stemming, n-grams and different BoW weighting options, all generated feature vectors are stored for later usage to speed up the experiment process.

3. Two Classifiers are tested, SVM and NB with as much as possible parameter configuration choices in order to reach an optimized parameter settings. The classifiers loop through all the generated feature vectors generated in the previous steps, results and trained models are stored.

4. Generated Models in the previous step are tested on unseen data, tests are conducted are same domain data and on different domain data, results are stored.

Figure 4: Experiment Phases

3.2 The Implementation

The implementation of the experiment was conducted in RapidMiner 7.2, the need for a tool that can rapidly build experiments was required due to the large number of datasets and different configuration settings required, implementing such an experiment using a programming language like

Python and the use of existing NLP package NLTK will require a long effort not needed. This section will explain each part of the designed experiment but without showing the details for each corpus, the details of each corpus experiment will be provided in an appendix.

3.2.1 Evaluation Metrics

The evaluation metrics in this research are mainly the use of F-Measure, which combines both precision and recall, the reason behind this is majority of the literature referenced here used it in reporting their results, yet still our experiments contains the details of f-measure and accuracy in all cases.

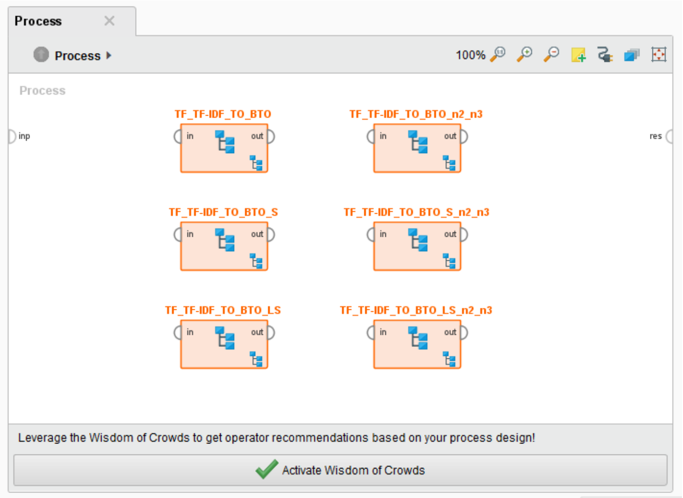
3.2.2 Feature Vector Generation

The main steps that are always done in the generation of the feature vectors are tokenization, filtering stop words and filtering words that are less than 3 characters. The other different options that have been used to generate different feature vectors are, stemming, light stemming, generation of n-grams and the document vector frequency weighting approach, Term Occurrence, Binary Term Occurrence, Term Frequency, and Term Frequency Inverse Document Frequency. Based on that, the different possible variation of feature vectors that can be generated are:

Stemming: 3 options, Stemming, Light Stemming and No Stemming
N-Grams: 3 options, Bi-grams, Tri-grams and no n-grams.
Frequency Weighting: 4 options, TO, BTO, TF, and TF-IDF
Total variations for one corpus = 3 x 3 x 4 = 36
Total generated feature vectors = 13 x 36 = 468

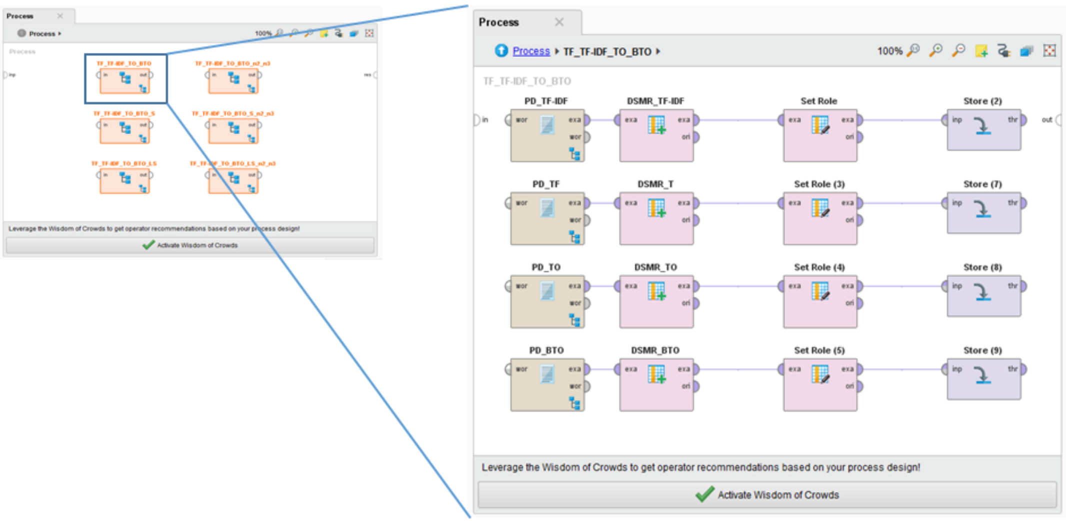
In order to speed up the processing, all possible variations were generated and stored for repetitive use. Figure 5 shows the design process in Rapid Miner for the DSMR dataset, each of these operators is a “subprocess” in which they contain other processes inside; it is basically a way of grouping a set of operators. The naming convention of each of these subprocess shows the different feature vector options it generates, for example, the first top-right subprocess is named TF_TF-IDF_TO_BTO, the underscore _ is a separator, So, we see here that this subprocess generated four feature vectors with different frequency weighting options 1)TF: Term Frequency 2) TF-IDF: Term Frequency Inverse Document Frequency 3) TO: Term Occurrence 4)BTO: Binary Term Occurrence, the details of this subprocess can be seen in figure 6. Another example TF_TF-IDF_TO_BTO_LS_n2_n3, the first four settings are same as before, LS refer to LightStemming, n2 refers to the use of bi-grams and n3 to tri-grams, this subprocess will generate eight feature vectors:

- TF_LS_n2
- TF-IDF_LS_n2
- TO_LS_n2
- BTO_LS_n2
- TF_LS_n3
- TF-IDF_LS_n3
- TO_LS_n3
- BTO_LS_n3



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

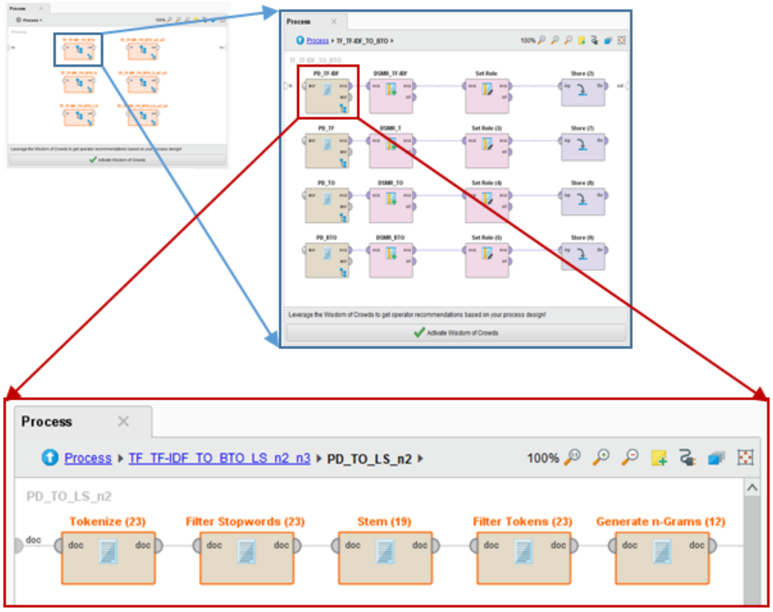
Figure 5: Main Document Process for the complete corpus



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 6: Details of the Sub-Processes TF_TF-IDF_TO_BTO

Figure 6 as explained shows the details of the first subprocess that generates four different feature vectors, the subprocess has four sets of operators each responsible for producing a feature vector, the first operator is a document process operator is the one that creates the feature vector, other operators basically are helper operators that assist in shaping and saving the feature vector, in figure 7 we can see the details of the document process operator, in this case, we see five operators, tokenize, filter stop words, stem (light stemming), filter tokens based on length and lastly generate n-grams(bi-gram).

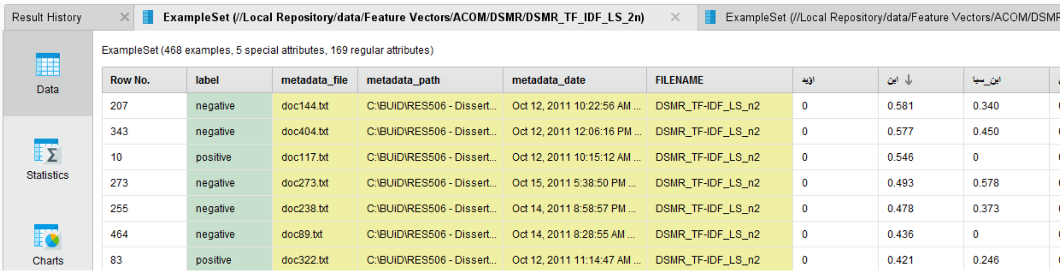


Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 7: Document Process Operator Details

The purpose of the other helper operators, the “Generate Attribute” and “Set Role”, the first was used to insert a new column containing a coded string of the feature vector settings, this was used later in the results analysis as a way to group results by the generated feature vector type in an automated way. The “Set Role” was required in order to identify the role of the newly generated column and not to include it in the training process.

Figure 8. shows the content of the generated feature vector (document matrix), the highlighted columns are special attributes and not included in the training process, we can see the label(polarity) of the document, the source file name, the file path, date and our generated attribute that shows the setting named FILENAME, in addition at the top the number of documents(referred to as examples) is mentioned with the number of special attributes and the number of words in the vocabulary, in this figure it mentions 468 examples(documents/reviews), 5 special attributes and 169 regular attributes(words).

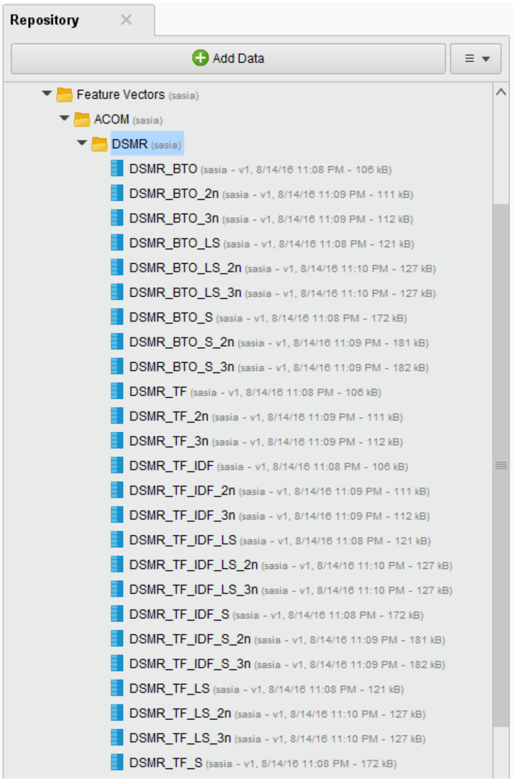

Result History
ExampleSet (//Local Repository/data/Feature Vectors/ACOM/DSMR/DSMR_TF_IDF_LS_2n)
ExampleSet (//Local Repository/data/Feature Vectors/ACOM/DSMR
Data
Statistics
Charts

ExampleSet (468 examples, 5 special attributes, 169 regular attributes)

Row No.	label	metadata_file	metadata_path	metadata_date	FILENAME	اؤيد	ابن ↓	ابن_سيا
207	negative	doc144.txt	C:\BUiD\RES506 - Dissert...	Oct 12, 2011 10:22:56 AM ...	DSMR_TF-IDF_LS_n2	0	0.581	0.340
343	negative	doc404.txt	C:\BUiD\RES506 - Dissert...	Oct 12, 2011 12:06:16 PM ...	DSMR_TF-IDF_LS_n2	0	0.577	0.450
10	positive	doc117.txt	C:\BUiD\RES506 - Dissert...	Oct 12, 2011 10:15:12 AM ...	DSMR_TF-IDF_LS_n2	0	0.546	0
273	negative	doc273.txt	C:\BUiD\RES506 - Dissert...	Oct 15, 2011 5:38:50 PM ...	DSMR_TF-IDF_LS_n2	0	0.493	0.578
255	negative	doc238.txt	C:\BUiD\RES506 - Dissert...	Oct 14, 2011 8:58:57 PM ...	DSMR_TF-IDF_LS_n2	0	0.478	0.373
464	negative	doc89.txt	C:\BUiD\RES506 - Dissert...	Oct 14, 2011 8:28:55 AM ...	DSMR_TF-IDF_LS_n2	0	0.436	0
83	positive	doc322.txt	C:\BUiD\RES506 - Dissert...	Oct 12, 2011 11:14:47 AM ...	DSMR_TF-IDF_LS_n2	0	0.421	0.246

Figure 8: Feature Vector content

All generated features vectors are stored in a specific folder in the RapidMiner repository; figure 9 shows the partial content of the DSMR corpus.



```
Feature Vectors (sasia)
  ACOM (sasia)
    DSMR (sasia)
      DSMR_BTO (sasia - v1, 8/14/16 11:08 PM - 106 kB)
      DSMR_BTO_2n (sasia - v1, 8/14/16 11:09 PM - 111 kB)
      DSMR_BTO_3n (sasia - v1, 8/14/16 11:09 PM - 112 kB)
      DSMR_BTO_LS (sasia - v1, 8/14/16 11:08 PM - 121 kB)
      DSMR_BTO_LS_2n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_BTO_LS_3n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_BTO_S (sasia - v1, 8/14/16 11:08 PM - 172 kB)
      DSMR_BTO_S_2n (sasia - v1, 8/14/16 11:09 PM - 181 kB)
      DSMR_BTO_S_3n (sasia - v1, 8/14/16 11:09 PM - 182 kB)
      DSMR_TF (sasia - v1, 8/14/16 11:08 PM - 106 kB)
      DSMR_TF_2n (sasia - v1, 8/14/16 11:09 PM - 111 kB)
      DSMR_TF_3n (sasia - v1, 8/14/16 11:09 PM - 112 kB)
      DSMR_TF_IDF (sasia - v1, 8/14/16 11:08 PM - 106 kB)
      DSMR_TF_IDF_2n (sasia - v1, 8/14/16 11:09 PM - 111 kB)
      DSMR_TF_IDF_3n (sasia - v1, 8/14/16 11:09 PM - 112 kB)
      DSMR_TF_IDF_LS (sasia - v1, 8/14/16 11:08 PM - 121 kB)
      DSMR_TF_IDF_LS_2n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_TF_IDF_LS_3n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_TF_IDF_S (sasia - v1, 8/14/16 11:08 PM - 172 kB)
      DSMR_TF_IDF_S_2n (sasia - v1, 8/14/16 11:09 PM - 181 kB)
      DSMR_TF_IDF_S_3n (sasia - v1, 8/14/16 11:09 PM - 182 kB)
      DSMR_TF_LS (sasia - v1, 8/14/16 11:08 PM - 121 kB)
      DSMR_TF_LS_2n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_TF_LS_3n (sasia - v1, 8/14/16 11:10 PM - 127 kB)
      DSMR_TF_S (sasia - v1, 8/14/16 11:08 PM - 172 kB)
```

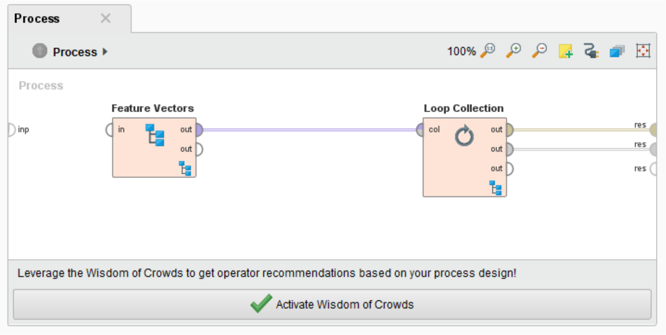
Figure 9: Feature Vector Folder Repository in RapidMiner

3.2.3 Training Classifiers

In this research, two classifiers were used the SVM and NB due to their popularity usage in solving SA. Here we include the details of the experiment design for part 1 where the trained models are optimized to produce the best possible set of parameter settings. The second part of the experiment design is simple training a classifier and requires no elaboration on the design as its straight forward. The description of design below is for those datasets that are file based, but for those where data is stored in a database is slightly different in how to retrieve the data other than that both are identical.

3.2.3.1 Design

In order to train the classifier with multiple features vectors in a reusable manner, a special operator was used to group all feature vectors into a collection, then through a loop operator each time a feature vector is chosen and feed into the classifier. Figure 10 shows the SVM/NB training classifier process; it contains two operators the first “Feature Vectors” is a subprocess operator where it groups all feature vectors into a collection as shown in figure 11.



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 10: SVM/NB Training Classifiers Process - First Level

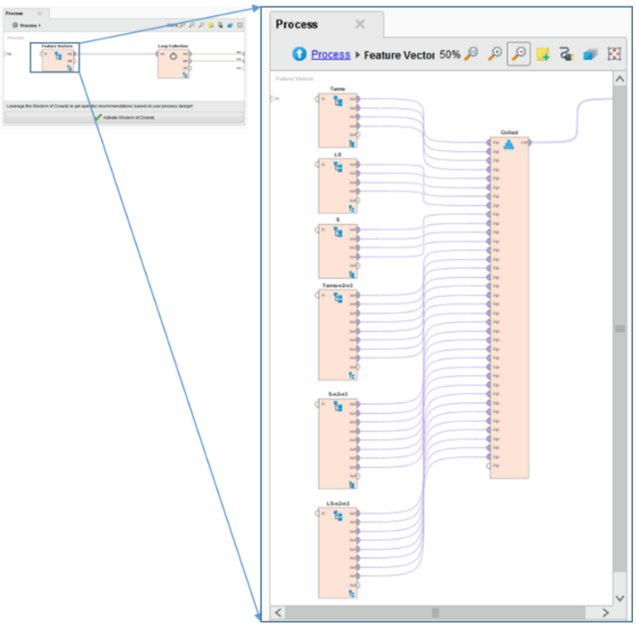


Figure 11: Feature Vector SubProcess Details

The Loop Collection operator iterates through the collection of feature vectors and repeats the processes inside it. Looking at figure 12, the loop collection operator contains 3 operators 1)OP: Optimize Operator, used to test the classifier with different settings, figure 13 shows the parameters configuration chosen, we can see two selected parameters, C, and kernel_type, C is configured with values 0,1 and 10, and Kernal_type with dot, radial, polynomial and anova, resulting in 12 different classifier combinations. Figure 14 shows the parameters configuration chosen for the

NB classifier, laplace correction, estimation mode and a number of kernels, a total of 44 different classifier combinations.

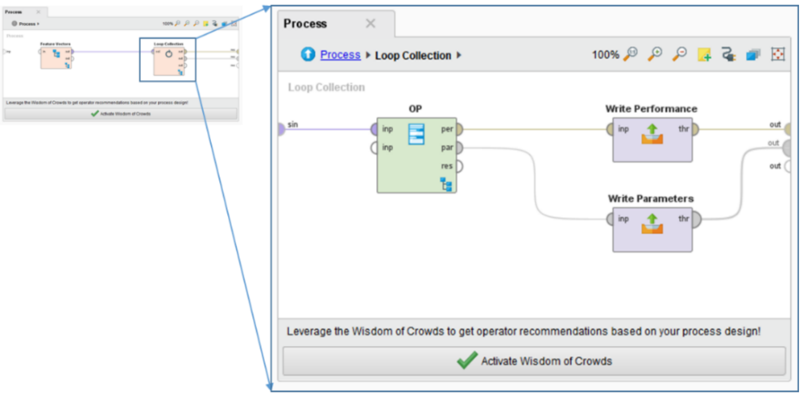
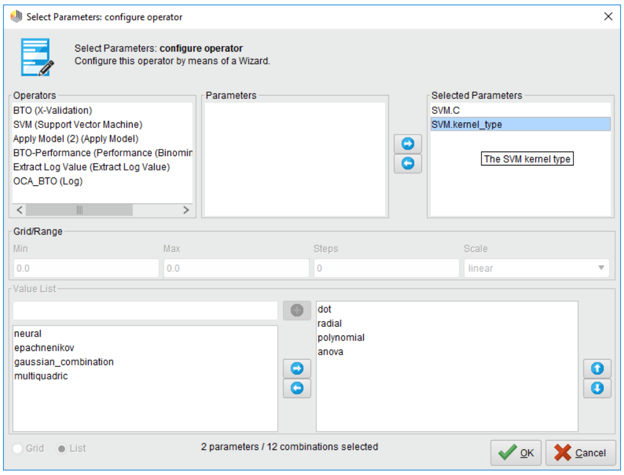

Process
Process ▸ Loop Collection ▸
100%
Loop Collection
OP
Write Performance
Write Parameters
sin
inp
inp
per
par
res
thr
out
out
out
Leverage the Wisdom of Crowds to get operator recommendations based on your process design!
Activate Wisdom of Crowds

Figure 12: The Loop Collection Operator Details – Second Level


Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators
BTO (X-Validation)
SVM (Support Vector Machine)
Apply Model (2) (Apply Model)
BTO-Performance (Performance (Binomir
Extract Log Value (Extract Log Value)
OCA_BTO (Log)

Parameters

Selected Parameters
SVM.C
SVM.kernel_type

The SVM kernel type

Grid/Range

Min	Max	Steps	Scale
0.0	0.0	0	linear

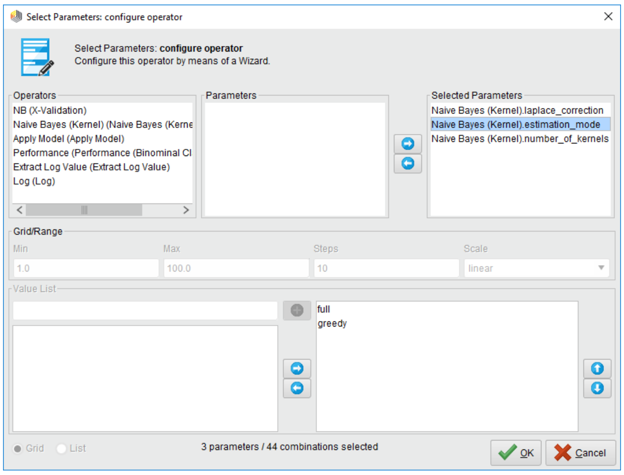
Value List
neural
epachnenikov
gaussian_combination
multiquadric

dot
radial
polynomial
anova

○ Grid ● List 2 parameters / 12 combinations selected

OK Cancel

Figure 13: The SVM Optimize Parameter Settings


Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators
NB (X-Validation)
Naive Bayes (Kernel) (Naive Bayes (Kerne
Apply Model (Apply Model)
Performance (Performance (Binominal Cl
Extract Log Value (Extract Log Value)
Log (Log)

Parameters

Selected Parameters
Naive Bayes (Kernel).laplace_correction
Naive Bayes (Kernel).estimation_mode
Naive Bayes (Kernel).number_of_kernels

Grid/Range

Min	Max	Steps	Scale
1.0	100.0	10	linear

Value List
full
greedy

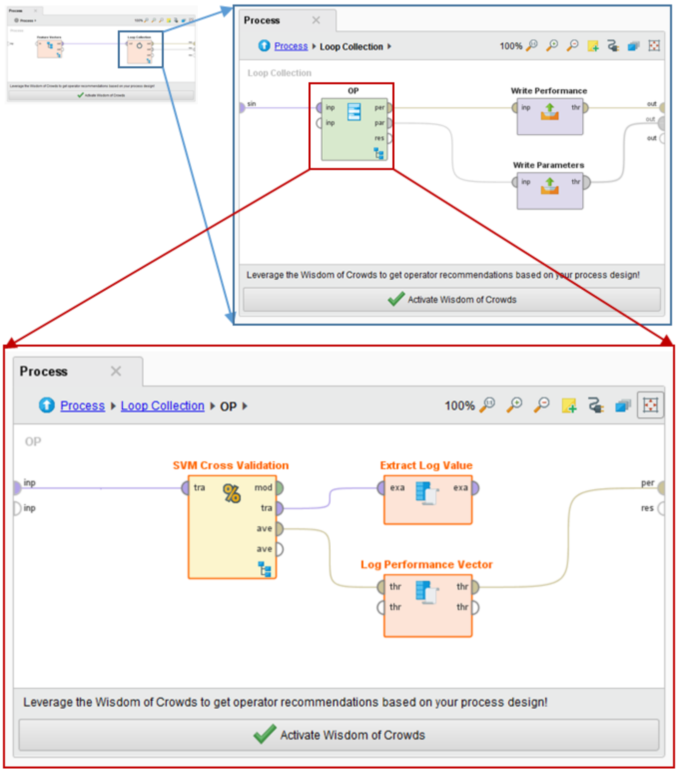
● Grid ○ List 3 parameters / 44 combinations selected

OK Cancel

Figure 14: The NB Optimize Parameter Settings

The laplace correction parameter is an optional parameter in RapidMiner implementation of Naïve Bayes Kernel; it is defined as follow “This parameter indicates if Laplace correction should be used to prevent high influence of zero probabilities. There is a simple trick to avoid zero probabilities. We can assume that our training set is so large that adding one to each count that we need would only make a negligible difference in the estimated probabilities, yet would avoid the case of zero probability values. This technique is known as Laplace correction.” (RapidMiner Help Documents)

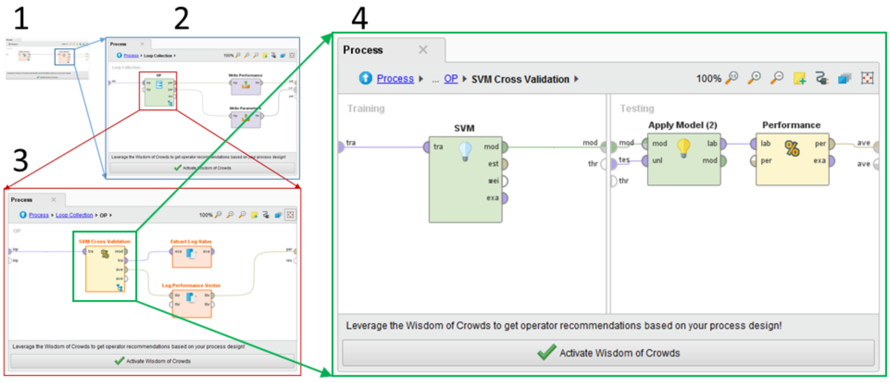
The optimize parameter operator contains 2 main operators as shown in figure 15, the SVM/NB cross-validation operator, and a log performance vector.



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 15: The SVM Optimize Parameter Operator Details – Third Level

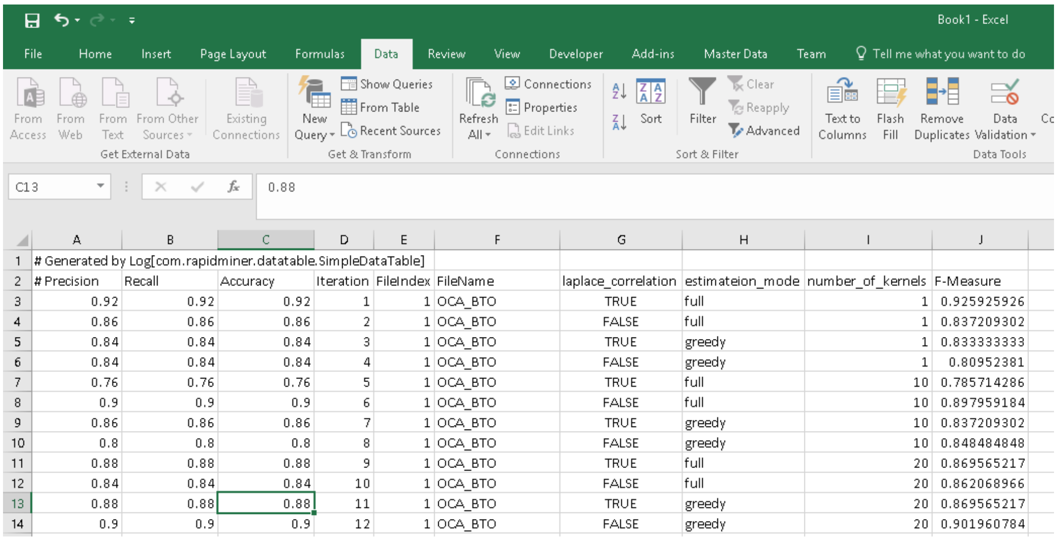


Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Figure 16: The SVM Cross Validation Operator Details - Fourth Level

3.2.3.2 Generated Output Files

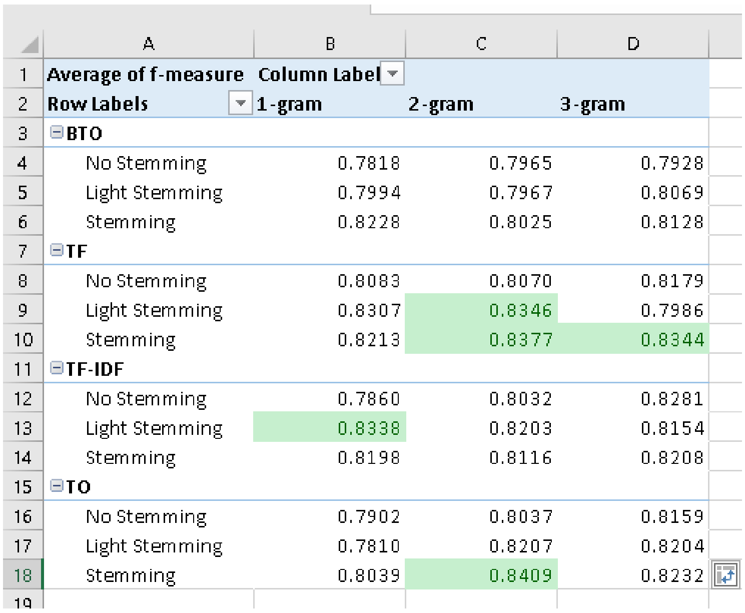
The training process generates three types of files, 1) Performance files: the basic output of the performance operator that displays the confusion matrix and reported accuracy, precision and recall values, in total for a dataset that means it will generate 36 files. 2) Parameter files: contains the best-optimized parameters found running the classifier on specific feature vector, in total for a dataset that means it will generate 36 files. 3) Performance log file: this is the main output of the process, it contains an accumulative result from all the tested feature vectors and all possible parameters combinations indicated in the experiment. For example, When training the OCA dataset using SVM classifier, we have 36 possible feature vectors, and 24 possible parameter combination to test, a total of 24x36 = 864 trained models were generated, and the performance is recorded, the file format is CSV and is used later for analysis. Figure 17 shows a partial rendering of such a file.



	A	B	C	D	E	F	G	H	I	J
1	# Generated by Log[com.rapidminer.datatable.SimpleDataTable]									
2	# Precision	Recall	Accuracy	Iteration	FileIndex	FileName	laplace_correlation	estimateion_mode	number_of_kernels	F-Measure
3	0.92	0.92	0.92	1	1	OCA_BTO	TRUE	full	1	0.925925926
4	0.86	0.86	0.86	2	1	OCA_BTO	FALSE	full	1	0.837209302
5	0.84	0.84	0.84	3	1	OCA_BTO	TRUE	greedy	1	0.833333333
6	0.84	0.84	0.84	4	1	OCA_BTO	FALSE	greedy	1	0.80952381
7	0.76	0.76	0.76	5	1	OCA_BTO	TRUE	full	10	0.785714286
8	0.9	0.9	0.9	6	1	OCA_BTO	FALSE	full	10	0.897959184
9	0.86	0.86	0.86	7	1	OCA_BTO	TRUE	greedy	10	0.837209302
10	0.8	0.8	0.8	8	1	OCA_BTO	FALSE	greedy	10	0.848484848
11	0.88	0.88	0.88	9	1	OCA_BTO	TRUE	full	20	0.869565217
12	0.84	0.84	0.84	10	1	OCA_BTO	FALSE	full	20	0.862068966
13	0.88	0.88	0.88	11	1	OCA_BTO	TRUE	greedy	20	0.869565217
14	0.9	0.9	0.9	12	1	OCA_BTO	FALSE	greedy	20	0.901960784

Figure 17: Performance Log File Sample

The file is later manipulated into generating a pivot table with up to 6 variables to investigate the behavior of the performance, figure 18 shows a sample of a pivot table based on the results of one of the experiments, the figure shows three variables, the term weight, stemming settings and chosen n-gram, values are the reported average f-measure and highlighted top 5 results.



	A	B	C	D
1	Average of f-measure	Column Label		
2	Row Labels	1-gram	2-gram	3-gram
3	⊟BTO			
4	No Stemming	0.7818	0.7965	0.7928
5	Light Stemming	0.7994	0.7967	0.8069
6	Stemming	0.8228	0.8025	0.8128
7	⊟TF			
8	No Stemming	0.8083	0.8070	0.8179
9	Light Stemming	0.8307	0.8346	0.7986
10	Stemming	0.8213	0.8377	0.8344
11	⊟TF-IDF			
12	No Stemming	0.7860	0.8032	0.8281
13	Light Stemming	0.8338	0.8203	0.8154
14	Stemming	0.8198	0.8116	0.8208
15	⊟TO			
16	No Stemming	0.7902	0.8037	0.8159
17	Light Stemming	0.7810	0.8207	0.8204
18	Stemming	0.8039	0.8409	0.8232
19				

Figure 18: Results Analysis using PivotTables

3.3 Conclusion

In this chapter, the methodology to the research approach is explained, along with the experiment design. The experiment design mainly falls into two parts, the first is responsible for generating the different possible feature vectors and store their results as an intermediate result to speed up the training process, the second part is the parameter optimization process itself. Also, the generated out files are explained specifically the log file the provides the necessary information to understand the reported performance.

Chapter 4

Results and Findings

In this chapter, the obtained resulted from the conducted experiments based on the methodology in chapter 3 are listed here and compared to the literature review results obtained, and answer to researcher questions are provided, the experiments are divided into three parts. Part1 is the evaluation of the best parameters settings to be used through parameter optimization the experiment is conducted on 4 datasets. Part 2, uses the findings from part 1 and tests and selected parameters on two new datasets. Part 3, attempts to evaluate the performance of a trained model on an unseen dataset from a different source.

4.1 Experiment Results Part 1

Here we detail the results of the conducted experiments on each corpus and compare obtained results to others found in the literature review that used the same dataset as shown in table 20.

4.1.1 OCA Corpus

The OCA corpus(Rushdi-Saleh et al. 2011), a movie reviews corpus containing 500 reviews split in half positive and negative, has been around for some time now and was tested in 4 papers including the authors of the corpus(Rushdi-Saleh et al. 2011) and the previous work for this research(Atia & Shaalan 2015), comparing their results to the obtained results from our experiment we see in most cases we achieved a higher performance by simply just implementing the parameter optimization methodology. The corpus was tested using two classifiers, the Naïve Bayes Kernel and the Support Vector Machine. We first show the results obtained from the Naïve Bayes and some findings then do the same with SVM classifier. In the end, we compare the best performance with results from the literature review.

Using the Naïve Bayes Kernel Classifier, a total of 864 model were trained, in table 22, the top 10 results of running a parameter optimization on Naïve Bayes classifier are shown, in overall the binary term occurrence weighting approach with n-grams provided the best results.

File Name	Laplace correction	Estimation mode	#kernels	Accuracy	F-Measure
OCA_BTO_S_n2	TRUE	Greedy	1.00	0.9800	0.9767
OCA_BTO_n2	FALSE	Full	10.00	0.9600	0.9655
OCA_TO	TRUE	Greedy	1.00	0.9592	0.9600
OCA_BTO_LS_n3	TRUE	Full	40.00	0.9600	0.9600
OCA_TF-IDF	TRUE	Greedy	1.00	0.9600	0.9583
OCA_BTO_S_n2	TRUE	Greedy	40.00	0.9600	0.9565
OCA_TO_n2	FALSE	Greedy	30.00	0.9556	0.9545
OCA_BTO_n3	TRUE	Greedy	50.00	0.9600	0.9500
OCA_BTO_n2	FALSE	Greedy	1.00	0.9400	0.9492
OCA_BTO_n3	TRUE	Greedy	1.00	0.9400	0.9492

Table 22:OCA-NB Classifier Top 10 Results

In table 23, the average f measure across six different variables, three related to feature vector generation and three to classifier parameters. The term weight is in favor to BTO as it was highlighted in the top 10 results, the bi-gram scored the highest average very close to trigram, stemming results are in general very close. The NB Kernel classifier is in favor to use Laplace correction, with greedy estimation mode and single kernel.

Term Weight	Avg. of F-Measure
BTO	**0.8468**
TF	0.7657
TF-IDF	0.7192
TO	0.7687

n-gram	Avg. of F-Measure
1-gram	0.7591
2-gram	**0.7851**
3-gram	0.7811

Stemming	Avg. of F-Measure
No Stemming	0.7760
Light Stemming	0.7702
Stemming	**0.7790**

Laplace Corr.	Avg. of F-Measure
FALSE	0.7715
TRUE	**0.7787**

Estimation Mode	Avg. of F-Measure
Full	0.7678
Greedy	**0.7824**

#Kernels	Avg. of F-Measure
1	**0.8129**
10	0.7592
20	0.7700
30	0.7801
40	0.7640
50	0.7643

Table 23: OCA-NB Parameters Average Score

Using the SVM classifier a total of 648 models were trained, in table 24, the top 10 results of running a parameter optimization on an SVM classifier are shown, in overall the Kernel type seems to be the only parameter that has actual effect on the results, as the feature vectors are of a different kind and the values of C are mainly 10 and 0.1.

FileName	# C	Kernal	Accuracy	F-Measure
OCA_TO_S	1000	Anova	1	1
OCA_TF-IDF_n3	0.1	Anova	1	1
OCA_TO_n2	0.1	Anova	1	1
OCA_TF-IDF_S_n3	10	Anova	1	1
OCA_TF-IDF_S_n3	0.1	Anova	1	1
OCA_TF_S_n2	10	anova	1	1
OCA_TO_S_n2	10	anova	1	1
OCA_TF_LS_n3	10	polynomial	1	1
OCA_TF_S_n3	0.1	anova	0.9696	0.9714
OCA_TF-IDF	10	anova	0.9696	0.9677

Table 24: OCA-SVM Classifier Top 10 Results

In table 25, the average f measure across five different variables, three related to feature vector generation and two to classifier parameters. The term weight is in favor to Term Frequency, the bi-gram scored the highest average very close to trigram, stemming results are in general very close. The SVM ANOVA kernel has the highest value as also shown from the top 10 results.

Term Weight	Avg. of F-Measure
BTO	0.7975
TF	**0.8589**
TF-IDF	0.8525
TO	0.7922

Kernel	Avg. of F-Measure
ANOVA	**0.8761**
dot	0.8039
polynomial	0.7959

n-gram	Avg. of F-Measure
1-gram	0.8140
2-gram	**0.8328**
3-gram	0.8291

C	Avg. of F-Measure
0.1	0.8236
10	**0.8328**
1000	0.8195

Stemming	Avg. of F-Measure
No Stemming	0.8227
Light Stemming	0.8232
Stemming	**0.8300**

Table 25: OCA-SVM Parameters Average Score

When comparing the results obtained with results from literature review as shown in table 26, obtained results from both classifiers topped results of previous work specifically (Amine Bayoudhi et al. 2015), as their ensemble hybrid approach should be a more advanced approach to solving the sentiment analysis problem.

Reference	F1	Approach
(Rushdi-Saleh et al. 2011)	0.91	Using SVM, TF-IDF, 3-gram
(Mountassir et al. 2013)	0.93	Using k-NN, SVM and NB reported lower results
(Amine Bayoudhi et al. 2015)	0.95	Ensemble-Based Classifier (Bagging, MaxEnt) + Multi-type feature set
Parameter Optimization	0.9767	Using NB, BTO
	1.00	Using SVM, ANOVA Kernel

Table 26: OCA Literature Review Results Comparison

4.1.2 ACOM Corpus

The ACOM corpus(Mountassir et al. 2013) contains three datasets, DSMR movie review, DSSP sports comments, DSPO political comments. The corpus, in general, has been used twice to the best we know, in the author original work and in (Amine Bayoudhi et al. 2015), our results will be compared to both works. Experiments are conducted in the same manner as reported on the OCA Corpus; two classifiers are used the NB and SVM, results for each classifier will be presented with some analysis of the best setting to be used and in the end a comparison of results.

4.1.2.1 ACOM-DSMR

The DSMR dataset contains 184 positive reviews and 284 negative reviews, the data is unbalanced and present a changed from the OCA corpus. Using the NB classifier a total of 1008 models were trained, the top 10 results of running a parameter optimization can be seen in table 27. same as in the OCA experiment, the binary term occurrence weighting approach with n-grams, provided the best results, giving us a first common setting to be used on NB classifiers.

File Name	Laplace correction	Estimation mode	#kernels	Accuracy	F-Measure
DSMR_BTO_LS	TRUE	full	50	0.8723	0.9063
DSMR_BTO_LS_n2	TRUE	full	1	0.8723	0.9063
DSMR_TO_S_n3	TRUE	full	20	0.8511	0.9041
DSMR_TF_LS_n3	FALSE	greedy	50	0.8511	0.9041
DSMR_BTO_LS_n2	TRUE	greedy	50	0.8723	0.9032
DSMR_BTO_S_n2	FALSE	greedy	10	0.8511	0.8889
DSMR_BTO_LS_n2	TRUE	greedy	30	0.8298	0.8857
DSMR_BTO_LS	TRUE	greedy	20	0.8511	0.8852
DSMR_TF_S_n2	TRUE	greedy	40	0.8298	0.8824
DSMR_BTO_LS	FALSE	full	30	0.8511	0.8814

Table 27:DSMR-NB Classifier Top 10 Results

In table 28 the average f1 score across all six variables are shown as done previously. Term weight, n-gram, and laplace correction settings are similar to the results obtained from the OCA NB experiment.

Term Weight	Avg. of F-Measure
BTO	**0.7709**
TF	0.7566
TF-IDF	0.7445
TO	0.7529

n-gram	Avg. of F-Measure
1-gram	0.7532
2-gram	**0.7579**
3-gram	0.7575

Stemming	Avg. of F-Measure
No Stemming	0.7243
Light Stemming	**0.7770**
Stemming	0.7673

Laplace Corr.	Avg. of F-Measure
FALSE	0.7558
TRUE	**0.7566**

Estimation Mode	Avg. of F-Measure
full	**0.7600**
greedy	0.7524

#Kernels	Avg. of F-Measure
1	0.7474
10	0.7580
20	0.7575
30	0.7565
40	**0.7603**
50	0.7585

Table 28: DSMR-NB Parameters Average Score

Using the SVM classifier, a total of 324 models were trained , in table 29, the top 10 results of running a parameter optimization on an SVM classifier are shown, in overall the Kernel type seems to be the only parameter that has actual effect on the results, as the feature vectors are of a different kind and the values of C are mainly 10 and 0.1.

FileName	# C	Kernal	Accuracy	f-Measure
DSMR_TF-IDF_S_n3	1000	polynomial	0.8085	0.8831
DSMR_TF-IDF_S_n3	10	polynomial	0.7872	0.8750
DSMR_TO_LS_n2	0.1	dot	0.8085	0.8732
DSMR_TF_S	1000	polynomial	0.8085	0.8615
DSMR_TO_LS	0.1	dot	0.8085	0.8525
DSMR_TO_LS_n2	0.1	polynomial	0.7872	0.8485
DSMR_TO_n3	10	anova	0.7447	0.8462
DSMR_TO_n3	0.1	anova	0.7872	0.8387
DSMR_TF-IDF_S_n3	1000	anova	0.7447	0.8333
DSMR_TF_LS_n2	10	polynomial	0.7234	0.8312

Table 29:DSMR-SVM Classifier Top 10 Results

In table 30 the average f1 score across all five variables are shown as done previously. Parameter settings for the SVM classifier here are different from what is obtained in the OCA Experiment.

Term Weight	Avg. of F-Measure
BTO	0.6841
TF	0.6984
TF-IDF	0.7070
TO	**0.7089**

n-gram	Avg. of F-Measure
1-gram	0.6808
2-gram	0.7041
3-gram	**0.7136**

Stemming	Avg. of F-Measure
No Stemming	0.6650
Light Stemming	0.7124
Stemming	**0.7210**

Kernel	Avg. of F-Measure
anova	0.6873
dot	0.7048
polynomial	**0.7067**

C	Avg. of F-Measure
0.1	**0.7255**
10	0.6814
1000	0.6916

Table 30:DSMR-SVM Parameters Average Score

When comparing the results obtained with results from literature review as shown in table 31, obtained results were better than reported results in (Mountassir et al. 2013), but fell behind the work reported in (Amine Bayoudhi et al. 2015), yet it reported better results if the discourse feature only was removed, were the f-measure dropped to 0.853 and that’s lower than our reported results even though they still used an ensemble-based classifier.

Reference	F1	Approach
(Mountassir et al. 2013)	0.875	Using NB
	≈ 0.775	Using SVM
(Amine Bayoudhi et al. 2015)	0.929	Ensemble-Based Classifier (Bagging, MaxEnt) + Multi-type feature set
	0.853	Ensemble-Based Classifier (Bagging, MaxEnt). No Discourse feature
Parameter Optimization	0.9063	Using NB, BTO
	0.8831	Using SVM, ANOVA Kernel

Table 31: DSMR Literature Review Results Comparison

4.1.2.2 ACOM-DSSP

The DSSP dataset is a sports comments corpus containing 486 positive comments and 517 negative comments. Using the Naïve Bayes Classifier, a total of 1584 models were trained (a different set of kernels were tested hence the increase in the number of trained models). The top 10 results are shown in table 32, and the parameters average score in table 33. The first consistent parameter from all 3 NB classifier experiments is the use of bi-grams and laplace correction, the bi-gram has been mentioned in literature review as a preferred setting as it captures enough temporal information (word sequence), the laplace correction as briefed in chapter 3 deals with high count of zero probabilities and tries to neutralize their effect, it is expected in such term-document matrix we would have this high volume of zero probabilities hence the correction works in favour for better results.

File Name	Laplace correlation	Estimation mode	#kernels	Accuracy	F-Measure
DSSP_BTO_LS_n2	TRUE	full	70	0.8852	0.9358
DSSP_TF-IDF_LS	FALSE	full	51	0.8852	0.9278
DSSP_TF-IDF_S_n3	FALSE	greedy	1	0.8689	0.9273
DSSP_TO_n3	FALSE	full	70	0.8689	0.9245
DSSP_TO_S	FALSE	greedy	100	0.8621	0.9231
DSSP_TF-IDF_S	TRUE	full	31	0.8525	0.9159
DSSP_TF-IDF_S_n2	FALSE	greedy	60	0.8525	0.9126
DSSP_TO_S	FALSE	greedy	60	0.8421	0.9091
DSSP_TF_S_n2	FALSE	greedy	31	0.8525	0.9091
DSSP_TF_S	TRUE	full	90	0.8361	0.9074

Table 32:DSSP-NB Classifier Top 10 Results

Term Weight	Avg. of F-Measure
BTO	0.7879
TF	**0.7910**
TF-IDF	0.7867
TO	0.7902

Laplace Correction	Avg. of F-Measure
FALSE	0.7887
TRUE	**0.7892**

Estimation Mode	Avg. of F-Measure
full	**0.7915**
greedy	0.7864

n-gram	Avg. of F-Measure
1-gram	0.7410
2-gram	**0.8150**
3-gram	0.8108

#Kernels	Avg. of F-Measure
1	0.7823
11	**0.7940**
21	0.7852
31	0.7881
41	0.7918
51	0.7875
60	0.7939
70	0.7916
80	0.7883
90	0.7857
100	0.7899

Stemming	Avg. of F-Measure
No Stemming	0.7299
Light Stemming	0.8145
Stemming	**0.8224**

Table 33:DSSP-NB Parameters Average Score

Using the SVM classifier, a total of 324 models were trained , in table 34, the top 10 results of running a parameter optimization on an SVM classifier are shown and in table 35 the parameters average f-measures.

FileName	C	Kernal	Accuracy	f-Measure
DSSP_T_LS_n2	0.1	polynomial	0.9180	0.9515
DSSP_T_LS	0.1	polynomial	0.9016	0.9464
DSSP_TF-IDF_LS_n2	1000	anova	0.8525	0.9204
DSSP_T_LS_n3	10	polynomial	0.8525	0.9159
DSSP_T_n2	10	polynomial	0.8361	0.9074
DSSP_T_LS_n2	10	anova	0.8361	0.9057
DSSP_BTO_LS	0.1	anova	0.8197	0.9009
DSSP_T_n2	0.1	polynomial	0.8197	0.8972
DSSP_TF-IDF_S_n2	0.1	anova	0.8197	0.8932
DSSP_TF-IDF_LS	10	anova	0.8033	0.8909

Table 34:DSSP-SVM Classifier Top 10 Results

Term Weight	Avg. of F-Measure
BTO	0.7423
TF	**0.7863**
TF-IDF	0.7694
TO	0.7656

n-gram	Avg. of F-Measure
1-gram	0.7049
2-gram	**0.7984**
3-gram	0.7963

Stemming	Avg. of F-Measure
No Stemming	0.6875
Light Stemming	0.7880
Stemming	**0.8235**

Kernel	Avg. of F-Measure
anova	0.7822
dot	0.7148
polynomial	**0.8024**

C	Avg. of F-Measure
0.1	**0.7754**
10	0.7634
1000	0.7597

Table 35: DSSP-SVM Parameters Average Score

When comparing the results obtained with results from literature review as shown in table 36, obtained results were better than reported results in (Mountassir et al. 2013), and in (Amine Bayoudhi et al. 2015) by a margin of almost 18%.

Reference	F1	Approach
(Mountassir et al. 2013)	≈ 0.77	Using NB
(Amine Bayoudhi et al. 2015)	0.806	Ensemble-Based Classifier (Bagging, MaxEnt) + Multi-type feature set
Parameter Optimization	0.9358	Using NB
	0.9515	Using SVM

Table 36:DSSP Literature Review Results Comparison

4.1.2.3 ACOM-DSPO

The final dataset in the ACOM corpus is the DSPO dataset that contains political comments; the dataset has been used only in (Amine Bayoudhi et al. 2015); it contains 149 positive documents and 462 negative documents. The used classifiers are applied in the same approach as in DSMR and DSSP, table 37 and 39 show the top 10 results applying the NB and SVM classifiers, tables 38 and 40 show the parameters average f-measure results.

File Name	Laplace correlation	Estimation mode	#kernels	Accuracy	F-Measure
DSPO_TF-IDF_S_n3	TRUE	greedy	1	0.8689	0.9286
DSPO_BTO_S	TRUE	greedy	40	0.8689	0.9231
DSPO_TF_S_n2	TRUE	greedy	30	0.8689	0.9216
DSPO_TF_S	FALSE	greedy	1	0.8525	0.9204
DSPO_TF-IDF_n3	TRUE	greedy	60	0.8525	0.9159
DSPO_TO_LS	TRUE	full	1	0.8525	0.9126
DSPO_TF-IDF_LS	TRUE	full	10	0.8525	0.9091
DSPO_BTO_S_n3	FALSE	full	30	0.8361	0.9074
DSPO_TF_n3	TRUE	greedy	30	0.8525	0.9072
DSPO_TF_S_n2	FALSE	greedy	30	0.8361	0.9057

Table 37: DSPO-NB Classifier Top 10 Results

Term Weight	Avg. of F-Measure
BTO	0.8014
TF	**0.8212**
TF-IDF	0.8154
TO	0.8111

n-gram	Avg. of F-Measure
1-gram	0.8066
2-gram	0.8146
3-gram	**0.8156**

Stemming	Avg. of F-Measure
No Stemming	0.8026
Light Stemming	0.8132
Stemming	**0.8210**

Laplace corr.	Avg. of F-Measure
FALSE	**0.8133**
TRUE	0.8113

Estimation Mode	Avg. of F-Measure
full	**0.8126**
greedy	0.8119

#Kernels	Avg. of F-Measure
1	0.8023
10	0.8144
20	0.8139
30	**0.8207**
40	0.8168
50	0.8053

Table 38:DSPO-NB Parameters Average Score

FileName	# C	Kernal	Accuracy	f-Measure
DSPO_TF-IDF_LS	10	polynomial	0.9016	0.9464
DSPO_TF_LS_n3	1000	anova	0.8852	0.9307
DSPO_TF-IDF_S_n3	0.1	polynomial	0.8689	0.9298
DSPO_TO_S_n3	0.1	dot	0.8689	0.9167
DSPO_TF_S_n2	10	polynomial	0.8361	0.9107
DSPO_TO_LS_n2	10	polynomial	0.8197	0.8991
DSPO_TF-IDF	10	polynomial	0.8197	0.8972
DSPO_TF_LS_n3	0.1	polynomial	0.8197	0.8972
DSPO_TF-IDF_n3	0.1	anova	0.8361	0.8958
DSPO_TF_LS	10	polynomial	0.8197	0.8952

Table 39: DSPO-SVM Classifier Top 10 Results

Term Weight	Avg. of F-Measure
BTO	0.7561
TF	0.7942
TF-IDF	**0.8103**
TO	0.7715

Kernel	Avg. of F-Measure
anova	0.8039
dot	0.7302
polynomial	**0.8154**

n-gram	Avg. of F-Measure
1-gram	0.7571
2-gram	**0.7972**
3-gram	0.7946

C	Avg. of F-Measure
0.1	0.7796
10	**0.7871**
1000	0.7823

Stemming	Avg. of F-Measure
No Stemming	0.7451
Light Stemming	0.7925
Stemming	**0.8114**

Table 40:DSPO-NB Parameters Average Score

When comparing the results obtained with results from literature review as shown in table 41, obtained results were better than reported results in (Amine Bayoudhi et al. 2015) by a margin of almost 4%.

Reference	F1	Approach
(Amine Bayoudhi et al. 2015)	0.903	Ensemble-Based Classifier (Majority Voting with SVM, MaxEnt, and ANN as base classifiers) + Multi-type feature set
Parameter Optimization	0.9286	Using NB
	0.9464	Using SVM

Table 41:DSPO Literature Review Results Comparison

4.1.3 Experiments Results Part 1 Summary

Based on the conducted experiments using two classifiers and four datasets, in table 42 will list the possible best parameter settings to be used using the NB classifier when used for Sentiment Analysis, and in table 43 the same for the SVM classifier.

DataSet	Term Weight	n-gram	Stemming	Laplace Correction	Estimation Mode	#Kernels
OCA	BTO	2-gram	Stemming	TRUE	greedy	1
DSMR	BTO	2-gram	Light Stemming	TRUE	full	NA
DSSP	TF	2-gram	Stemming	TRUE	full	NA
DSPO	TF	3-gram	Stemming	FALSE	full	NA
Overall	**BTO,TF**	**2-gram**	**Stemming**	**TRUE**	**full**	**NA**

Table 42: NB SA Classifier Optimized Parameters

DataSet	Term Weight	n-gram	Stemming	Kernel Type	C
OCA	TF	2-gram	Stemming	ANOVA	10
DSMR	TO	3-gram	Stemming	Polynomial	0.1
DSSP	TF	2-gram	Stemming	Polynomial	0.1
DSPO	TF-IDF	2-gram	Stemming	Polynomial	10
Overall	**TF**	**2-gram**	**Stemming**	**Polynomial**	**0.1, 10**

Table 43: SVM SA Classifier Optimized Parameters

In overall the conducted experiment showed that through parameter optimization higher levels of accuracy could be achieved using basic classifiers compared to more rich approaches as shown in table 44.

Dataset	Classifier	Performed Better?	%diff	Dataset size	Comments
OCA	SVM	Yes	5.0%	500	
DSMR	NB	No	-3.0%	468	the corpus size could be the reason why parameter optimization did not achieve a better performance
DSSP	SVM	Yes	15.0%	1003	
DSPO	SVM	Yes	4.3%	611	

Table 44: Summary Results Experiments Part 1

4.2 Experiment Results Part 2

In this section, the obtained findings from the previous section will be used to train models for new datasets with no parameter optimization and compare their performance to those found in literature review. The chosen corpus is the one provided by(ElSahar & El-Beltagy 2015), it contains five different domain datasets and has recently been published.

4.2.1 Mov Dataset

The Mov dataset is a movie review dataset containing 969 positive reviews and 384 negative reviews. The NB and SVM classifiers were used and tuned to the settings in obtained from the first part of the experiment. The results of the NB classifier are shown in table 45, only the feature vectors needed are tested, in this case, it will be four using BTO and TF with both stemming and light stemming. The top accuracy result recorded was

0.8222 compared to 0.743 in (ElSahar & El-Beltagy 2015) that used a hybrid approach for classification.

FileName	laplace correction	estimation mode	Accuracy	F-Measure
MOV_BTO_S_n2	TRUE	full	0.8222	0.6364
MOV_TF_S_n2	TRUE	full	0.7778	0.5000

Table 45: MOV-NB Classifier Results

Testing the SVM classifier did not produce good f1 measure results, but on the accuracy level a value of 0.8296 was recorded compared to the same reported results by (ElSahar & El-Beltagy 2015) of 0.743 show a much better performance using a basic classifier with predefined parameters settings.

FileName	C	Kernal	Accuracy	F-Measure
MOV_TF_S_n2	0.1	polynomial	0.7926	0.4815
MOV_TF_S_n2	10	polynomial	0.8296	0.549

Table 46: MOV-SVM Classifier Results

4.2.2 HTL Dataset

The HTL dataset is a hotel's reviews data containing 10775 positive reviews and 2647 negative ones. The authors reported a top accuracy level of 0.887, in tables 47 and 48 we list our results of training an NB and SVM classifiers on the dataset. In 3 out 4 cases we report higher results by a margin up to 4%.

FileName	laplace_correlation	estimateion_mode	Accuracy	F-Measure
HTL_BTO_S_n2	TRUE	full	0.8636	0.663
HTL_TF_S_n2	TRUE	full	0.9039	0.6993

Table 47:HTL-NB Classifier Results

FileName	C	Kernal	Accuracy	F-Measure
HTL_TF_S_n2	0.1	polynomial	0.9218	0.7661
HTL_TF_S_n2	10	polynomial	0.9210	0.7782

Table 48: HTL-SVM Classifier Results

4.3 Experiment Results Part 3

In this section of the experiments we evaluate is it possible to use a pre-trained model on an unseen dataset from a different data source but in the same domain. It is possible but not in a straight forward process as it requires after preprocessing the unseen dataset to be filtered to only the attributes that were used by the original model, causing the feature vector generated to be further reduced losing important feature creates created in the feature vector generation.

This is due this research approach on how to represent feature vectors and most probably in all literature review that doesn't convert the feature vector to an abstract representation instead of words will not be able too.

The trained model was based on a set of feature vectors composed of the dataset, and after some pre-processing the number of attributes (words) are reduced to a subset that matter the most, causing the vocabulary size of the model to be only limited to that vocabulary, when the model is used against a new unseen dataset and from a different source, even if it is in the same domain it will most probably face unrecognizable attributes. Hence the model will fail. Even if the model was trained a huge corpus, there would still remain a chance that the model will not be able to recognize a new word. The resolution to have a generalized model is that the feature vectors used in training need to be transformed into abstract attributes, one possible way is to use a word embedding technique like word2vec.

4.4 The Answers to Research Questions

Here answers to the research questions are provided based on the obtained results.

4.4.1 RQ1

Q. Is it possible through parameter optimization and a basic classifier achieve higher levels of performance compared to other more rich and hybrid approaches?

A. Yes, through the first part of the experiments, four dataset were experimented with by performing a parameter optimization grid search on a specific set of parameters. In 3 of 4 cases, accuracy levels in terms of f-measure were higher than what was found in the literature review as shown in table 44.

4.4.2 RQ2

Q. Can we establish a predefined parameter configuration based on parameter optimization for both feature vector generation and classifier configuration to be used on other dataset and achieve good results?

A. Yes, as a result of the experiments from part one, a predefined parameter configuration was established for both NB and SVM classifiers as shown in tables 42 and 43. Using the results and experimenting with two new datasets we achieved better results than what was published.

4.4.3 RQ3

Q. Can a classification model built using a domain specific sentiment corpus achieve comparable results on a blind data set in the same domain?

A. No, as explained in section 4.3 part 3 of the experiment, the model were trained on a specific subset of the original vocabulary dataset, making the model hard-wired to that dataset and not able to process any new corpus even if it is in the same domain, as it is not possible to account for all possible words in the. This requires to train the models on an abstracted feature vector using either lexicon-based approach where only a predefined list of sentiment lexicon are captured or in a more generalized approach using word embedding were word are converted into numerical vector representations.

4.5 Conclusion

In this chapter the experiment results were presented with coparision to work in the literature review, through parameter optimization higher levels of performance were achieved using basic classifiers and simple feature generation process compared to other more elaborate work in the field on the same tested dataset. In addition it was possible to establish a predefined list of parameter settings for both feature vector generation and classifier settings and be used on the new dataset and still achieve good results compared to the literature review. We concluded the chapter by answering the three research questions presented in chapter one.

Chapter 5

Enhancements

The conducted investigation showed good results in establishing a base setting of parameters for both feature vector generation and classifier settings. The main enhancement part would be focusing on the feature vector generation as no advanced approach was used in the feature engineering process.

Feature engineering is probably a cornerstone when using machine learning, but yet it hasn’t had it is enough share of focus, it is likely due to the fact its more of an art than a procedure to follow as defined in (Deshpande 2016) feature engineering is “the art and science of selecting and/or generating the columns in a data table for a machine learning model”, the author explains that feature engineering is divided into three categories, feature selection, dimension reduction and feature generation.

Feature selection will be the process where the attributes are ranked based on importance, weight or any other criteria and based on a threshold value a selection is made, to relate to our research, attribute were selected based on their frequency in documents and based on prune values above and below certain value attributes/words are dropped and not included in feature vector.

Dimension reduction as the name implies deals with reducing the high dimensionality of a feature vector to a size that can be computationally possible, Principle Component Analysis is one technique that can be used if required. Lastly feature generation is where new attributes are generated to describe the dataset, again to relate to our case the use of n-grams is a feature generation act. For each of these three steps below are the possible ways that can further enhance this research.

5.1.1 Feature Selection

This research used a basic prune approach to select the attributes of focus, in our case the attributes are actually words, and in a problem as SA, cutting off the wrong adjective might cause an incorrect classification. In order to avoid this, a sentiment lexicon needs to be used to detect all sentiment words in the feature vector before any cutting off.

5.1.2 Dimension Reduction

No Dimensionality reduction was performed in this research, and in some cases, feature vectors were around 1000 attributes, but the classifiers managed to get trained on such vector size. There was an issue with one of the large datasets the LABR(Nabil et al. 2014) were it contained around 60K documents, the result feature vector was too large to be trained without performing a dimensionality reduction, as the process required some further investigation on how this might affect the results a different dataset was chosen.

5.1.3 Feature Generation

Although n-grams were generated in this research, other linguistic feature can be generated in order to provide a more informative feature vector that can aid in increasing the classification accuracy, most of the research focus on extracting more linguistic feature as it believed it is the approach to enhance the SA process further.

Chapter 6

Conclusion and Future Work

In this chapter, a conclusion to the research is presented along with the proposed future work.

6.1 Conclusion

The sentiment analysis problem contains several challenging tasks of which this research focuses on sentiment classification, several approaches exist, machine learning based, lexicon based or hybrid approaches. The research in Arabic language suffers from the lack of enough linguistics resources, causing on impact on research where several researchers start their work by building their own corpus or sentiment lexicon before approaching the sentiment classification problem, this has had an effect when researchers try to compare their results to others work, the solution would be for the author himself to establish his own baseline or benchmark results and use that as a measure to his work. Usually these benchmark or baseline results are generated using machine learning classifiers and in general SVM and NB have been widely accepted by all as they provide a good result in almost not time. In this research we have attempted to explore further the possibility of what can be achieved with these classifiers through parameter optimization, it is a well know practice of training a model, further tuning can be established by playing around with the classifiers parameters, but to the best of our knowledge no sufficient resource was found on how to fine tune these classifiers before starting! It is true through literature review specific settings are reported to be better in some cases and other in some other, for example the TF-IDF is widely used in text mining and usually it’s the first thing to use, but in sentiment classifciaiton indicated that TF-IDF performed lower in comparision to BTO and TF, for BTO or binary term occurrence it was mentioned that what is important is sentiment classification is the occurrence of the word, not how many times it occurs. The fine tuning is usually domain specific, So, in this research, an experiment was designed to test four different datasets from various domains and perform a parameter optimization in order to reach a conclusion of what would be the best possible set of parameters to be used on these classifiers to solve a sentiment classification problem. The investigation also included the part of the generation of feature vectors, where usually the focus goes to generating features based on a linguistic characteristic or the use of advanced sentiment lexicons, in this research, a focuses on the main pre-processing, and linguistic features were investigated. Chapter 2 provided the necessary literature review of the subject in general and focused on Arabic language, chapter 3 explained the methodology and the experiment design, chapter 4 reported the results and answers to our presented research questions in chapter 1. Through the first part of the experiment an overall all parameter setting and configuration, choices were detected as shown in table

	Term Weight	n-gram	Stemming	Laplace Correction	Estimation Mode
NB	BTO, TF	2-gram	Stemming	TRUE	full

	Term Weight	n-gram	Stemming	Kernel Type	C
SVM	TF	2-gram	Stemming	Polynomial	0.1, 10

Table 49: Optimized parameter Settings for SVM and NB

The optimized classifiers also outperformed other work in the literature review which used more advanced approached in solving the sentiment classification problem either by using advanced linguistic features a hybrid classifier approach. Table 50 summarizes the obtained results from the conducted experiment

Dataset	Classifier	Performed Better?	%diff
OCA	SVM	Yes	5.0%
DSMR	NB	No	-3.0%
DSSP	SVM	Yes	15.0%
DSPO	SVM	Yes	4.3%

Table 50: Results Comparison to Literature Review

The optimized parameters were also tested in a new scenario using two unseen datasets, in both cases reported results were better than what was published in the literature. The final part of the research was aimed to explore testing a trained model on unseen dataset from a different source in hope to generalize the model, but unfortunately due to the feature vector design approach it would reduce the performance dramatically, as the optimized models were trained with a feature vector with an attribute list based on the trained corpus, the attribute list is the word list and n-gram list, no matter what was the corpus size it is not possible to account for all possible words, the solution is to filter the generated feature vector from the new data set to only the list of attributes that have been previously trained in model. The solution as explained would be to transform the feature vector into an abstract representation of attribute (no longer represented as words) using a word embedding technique like word2vec, such approaches are used in deep learning and are discussed in the following section.

In overall the research has achieved its goal and provided some new guidelines on how to fine tune ML classifiers before training when solving a sentiment classification problem. Also, it is possible through this optimization that when applying any of the techniques found in the literature performance will increase.

6.2 Future Work

Here were present what would be the future work for this research. In chapter 5 a list of enhancement was mentioned that would improve the finding of this study, but as for future work the desired approach should be the use of Deep Learning (DL). The literature review showed to the best of our knowledge only one attempt in solving the SA classification in Arabic using Deep Learning(Sallab et al. 2015), the authors propose four different deep learning architecture, based on Deep Belief Networks(DBN), Deep Auto-Encoders (DAE), and Recursive Auto-Encoder (RAE). The research used the LDC Arabic Tree Bank dataset. The evaluation was compared to other state-of-the-art sentiment classification systems using the same dataset. Their Recursive Auto-Encoder recorded the highest results.

According to (Tang et al. 2015) the field of sentiment analysis is an active topic. The challenges in feature engineering require finding methods that can solve the problem without the extensive work required. The paper discusses the successful approaches recently used in the field, with a focus on the different sentiment analysis tasks, word embedding, sentiment classification, opinion extraction and lexicon sentiment learning.

Deep Learning is at its core a neural network with multiple hidden layers, So, the building blocks that make up a neural network are the same in deep learning. Neural Networks are used to address several problems; each type has its specific capabilities that make's it superior in solving its problems in (Melorose et al. 2015) listed in table 51 the different types of neural networks and what domain problem they solve with a level of degree indicated by the number of checkmarks. The ones of interest to address the sentiment classification problem have been highlighted.

	Clust	**Regis**	**Classif**	**Predict**	**Robot**	**Vision**	**Optim**
Self-Organizing Map	✓✓✓				✓	✓	
Feedforward		✓✓✓	✓✓✓	✓✓	✓✓	✓✓	
Hopfield			✓			✓	✓
Boltzmann Machine			✓				✓✓
Deep Belief Network			✓✓✓		✓✓	✓✓	
Deep Feedforward		✓✓✓	✓✓✓	✓✓	✓✓✓	✓✓	
NEAT		✓✓	✓✓		✓✓		
CPPN					✓✓✓	✓✓	
HyperNEAT		✓✓	✓✓		✓✓✓	✓✓	
Convolutional Network		✓	✓✓✓		✓✓✓	✓✓✓	
Elman Network		✓✓	✓✓	✓✓✓			
Jordan Network		✓✓	✓✓	✓✓	✓✓		
Recurrent Network		✓✓	✓✓	✓✓✓	✓✓	✓	

Table 51: Neural Network Types & Problem Domains

(Adapted from Melorose et al. 2015, p. 23)

The different problem domains are Clustering (Clust), Regression (Regis), Classification(Classif), Prediction(Predict), Robotics (Robot), Computer Vision(vision) and Optimization (Optim). The sentiment analysis problem is considered a classification problem where we need to classify a document or sentence to a polarity. From table 51 and based on the number of checkmarks the best possible used neural network types for the classification problem have been highlighted, Feedforward(FF), Deep Belief Network(DBN), Deep Feedforward(DFF) and Convolutional Network (CN), and we can see that Hopfield, Boltzmann Machine perform poorly on such type of problem. Several tools are currently available to train deep networks; table 52 list some of these tools.

Programming Tools for DNN
Theano
Torch
Caffe
Neon
Tensor Flow
DeepLearning 4j
H2O
Mxnet
Veles
Kaldi
SparkMLLIB
Kersa
SINGA
Accord.NET
SciKit

Table 52: Programming Tools for DNN

These type of network require presenting the attributes(words) in an abstract numerical format; word embedding is used to transform the corpus into a numerical representation, one technique known as word2vec transforms each word into a real numeric vector in space, such transformation presented interesting results, probably the most famous example goes as follow, King – man + woman = Queen. The transformation of words into a real-numeric vector space allowed such mathematical intuitive result to be established. The process itself of word embedding is followed by other steps before training deep networks, but it can be seen here such opportunities in using such approach.

References

Abdul-Mageed, M. & Diab, M., 2014. SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis. *Proceedings of the Language Resources and Evaluation Conference*, pp.1162–1169. Available at: http://www.lrec-conf.org/proceedings/lrec2014/pdf/919_Paper.pdf.

Abdul-Mageed, M., Diab, M. & K??bler, S., 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech and Language*, 28(1), pp.20–37. Available at: http://dx.doi.org/10.1016/j.csl.2013.03.001.

Al-ayyoub, M. et al., 2016. Evaluating SentiStrength for Arabic Sentiment Analysis Evaluating SentiStrength for Arabic Sentiment Analysis. , (July).

Al-ayyoub, M. & Nuseir, A., 2016. Hierarchical Classifiers for Multi-Way Sentiment Analysis of Arabic Reviews. , 7(2), pp.531–539.

Atia, S. & Shaalan, K., 2015. Increasing the Accuracy of Opinion Mining in Arabic. *The International Conference on Arabic Computational Linguistics (ACLing)*. Available at: http://dx.doi.org/10.1109/ACLing.2015.22.

Badaro, G., Baly, R. & Hajj, H., 2014. A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining. *Arabic Natural Language Processing Workshop co-located with EMNLP 2014, Doha, Qatar*, pp.176–184.

Bayoudhi, A. et al., 2015. Sentiment Classification At Discourse Segment Level: Experiments on multi-domain Arabic corpus. *Document-level school lesson …*, 30(1), pp.1–25. Available at: http://www.jlcl.org/2015_Heft1/1Bayoudhi.pdf.

Bayoudhi, A., Belguith, L. & Ghorbel, H., 2015. Sentiment Classification of Arabic Documents : Experiments with multi-type features and ensemble algorithms. , pp.196–205. Available at: http://bcmi.sjtu.edu.cn/~paclic29/proceedings/PACLIC29-1023.206.pdf.

Cambria, E. et al., 2013. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, 28(2), pp.15–21. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6468032.

Cambria, E., Livingstone, A. & Hussain, A., 2012. The hourglass of emotions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7403 LNCS, pp.144–157.

Deshpande, B., 2016. Learning data science: feature engineering. Available at: http://www.simafore.com/blog/learning-data-science-feature-engineering [Accessed August 15, 2016].

DOMO, 2016. Data Never Sleeps 2.0. *Report*. Available at: https://www.domo.com/learn/data-never-sleeps-2 [Accessed August 28, 2016].

Elarnaoty, M., AbdelRahman, S. & Fahmy, A., 2012. A Machine Learning Approach for Opinion Holder Extraction in Arabic Language. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 3(2), pp.45–63. Available at: http://arxiv.org/abs/1206.1011.

ElSahar, H. & El-Beltagy, S.R., 2015. Building Large Arabic Multi-domain Resources for Sentiment Analysis. In A. Gelbukh, ed. *Florida dental journal*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 23–34. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-84858326059&partnerID=tZOtx3y1.

Ganeshbhai, S.Y. & Shah, B.K., 2015. Feature based opinion mining: A survey. *Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015*, pp.919–923.

Glorot, X., Bordes, A. & Bengio, Y., 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. *Proceedings of the 28th International Conference on Machine Learning*, (1), pp.513–520. Available at: http://www.icml-2011.org/papers/342_icmlpaper.pdf.

Irsoy, O. & Cardie, C., 2014. Opinion mining with deep recurrent neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.720–728. Available at: http://anthology.aclweb.org/D/D14/D14-1080.pdf\nhttp://www.aclweb.org/anthology/D14-1080.

Kim, S.S.-M.S. et al., 2004. Determining the sentiment of opinions. *Proceedings of the 20th international conference on …*, p.1367–es. Available at: http://portal.acm.org/citation.cfm?doid=1220355.1220555\nhttp://dl.acm.org/citation.cfm?id=1220555.

Liu, B., 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), pp.1–167. Available at: http://www.morganclaypool.com/doi/abs/10.2200/S00416ED1V01Y201204HLT016.

Liu, B., 2010. Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*, pp.1–38. Available at: http://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf\nhttp://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment Analysis and Subjectivity-NLPHandbook-2010.pdf.

Liu, B. & Zhang, L., 2012. A survey of opinion mining and sentiment analysis. *Mining Text Data*, 9781461432, pp.415–463.

Liu, P., Joty, S. & Meng, H., 2015. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September), pp.1433–1443. Available at: http://aclweb.org/anthology/D15-1168.

Melorose, J., Perroy, R. & Careas, S., 2015. *Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning*,

Moraes, R., Valiati, J.F. & Gavião Neto, W.P., 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, 40(2), pp.621–633.

Mountassir, A., Benbrahim, H. & Berraba, I., 2013. Sentiment classification on arabic corpora. A preliminary cross-study. *Document numérique*, 16(1), pp.73–96. Available at: http://www.scopus.com/inward/record.url?eid=2-s2.0-

84879022187&partnerID=tZOtx3y1.

Nabil, M., Aly, M. & Atiya, A., 2015. ASTD: Arabic Sentiment Tweets Dataset. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September), pp.2515–2519. Available at: http://aclweb.org/anthology/D15-1299.

Nabil, M., Aly, M. & Atiya, A., 2014. LABR: A Large Scale Arabic Sentiment Analysis Benchmark. *Aclweb.Org*, pp.1–10. Available at: http://arxiv.org/abs/1411.6718.

OMA-Project, 2016. ArSenL Arabic Sentiment Lexicon Web Interfce. Available at: http://oma-project.com/ArSenL.

Rushdi-Saleh, M. et al., 2011. OCA: Opinion corpus for Arabic. *Journal of the American Society for Information Science and Technology*, 62(10), pp.2045–2054. Available at: http://arxiv.org/abs/0803.1716.

Sallab, A.A. Al, Baly, R. & Hajj, H., 2015. *Deep Learning Models for Sentiment Analysis in Arabic*, Available at: http://www.aclweb.org/anthology/W15-32#page=21.

Al Shboul, B., Al-Ayyouby, M. & Jararwehy, Y., 2015. Multi-way sentiment classification of Arabic reviews. *2015 6th International Conference on Information and Communication Systems, ICICS 2015*, (April 2015), pp.206–211.

Stanford, 2016. Stanford CoreNLP V3.6.0 Supported Languages. Available at: http://stanfordnlp.github.io/CoreNLP/ [Accessed August 21, 2016].

Tang, D., Qin, B. & Liu, T., 2015. Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6), pp.292–303.

Thelwall, M., 2013. Heart and soul: Sentiment strength detection in the social web with sentistrength. *Proceedings of the CyberEmotions*, 5, pp.1–14.