PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA

Faculty of Mathematics and Computer Science

Computer Science Department

N° :……..

DOMAINE : Mathematics and Computer Science

BRANCH : Computer science

SPECIALTY: Artificial Intelligence

Presented for obtaining the diploma of
Master in: COMPUTER SCIENCE

BY: Djakam Kaddour

Madani Okba

Title of thesis

Sentiment analysis for Arabic multi-dialects using hybrid approach

Composition of the jury :

DR. Bentercia Rahima	University of M'sila	Supervision
DR. Brahimi Mahmoud	University of M'sila	President
MR. Bougherara Seddik	University of M'sila	Examiner

Academic year: 2022 / 2023

DEDICATION

I would like to gift this humble work to : those who brought me to this rank of science and who spent sleepless nights for me, Allah the Exalted said:
« **واخفض لَهُمَا جَنَاحَ الذُّلِّ مِنَ الرَّحْمَةِ وَقُل رَّبِّ ارْحَمْهُمَا كَمَا رَبَّيَانِي صَغِيرًا** » El israa - 24
My Dear Parents To my beloved mother, you were my guiding light, my source of strength, and my biggest cheerleader. Your nurturing nature and endless sacrifices shaped me into the person I am today. Your gentle encouragement and belief in my dreams gave me the courage to pursue my passions. Although you are no longer here, your spirit lives on within me, inspiring me to embrace life with compassion and resilience. This work is dedicated to you, my dear mother, as a heartfelt tribute to the immeasurable impact you had on my life. I carry your love with me always.

My Dear father, who held everything for us.

My Siblings, The ones whom giving too much joy to life.

MADANI OKBA

I would like to gift this humble work to: my loving family, who has always supported and believed in me, thank you for being my constant source of strength and inspiration throughout this journey.

DJAKAM KADOUR

ACKNOWLEDGMENTS

First and above all, we thank “ALLAH” for giving us the patience, the health, the courage to have come this far.

We would like to thank **Dr. Rahima Bentrcia** who honored us with her supervision, for her direction, her Guidance, her advice and all her constructive remarksforthe good progress of our work.

ملخص

الهدف من بحثنا هو تحليل المشاعر المعبر عنها باللغة العربية ولهجاتها، في مجال السفر بعد استخلاص المشاعر المتعلقة بالسفر من النصوص، وتحليلها بدقة وفقا للهجات العربية. من خلال تطوير نهجين، الاول يعتمد على الانطولوجيا. والثاني يجمع بين الانطولوجيا والتعلم الالي. لتحليل النصوص بخمسة وعشرين لهجة مختلفة.

الكلمات المفتاحية:
تحليل المشاعر، تحليل المشاعر المعبر عنها بالعربية، اللهجات العربية، الانطولوجيا، التعلم .الالي

Abstract

The objective of our research is to analyze the feelings expressed in the Arabic language and its dialects in the field of travel. This involves extracting travel-related emotions from texts and analyzing them accurately according to the various Arabic dialects. We have developed two approaches for this purpose. The first approach relies on ontology, while the second approach combines ontology with machine learning. These approaches are used to analyze texts in 25 different dialects.

Keywords: sentiment analysis, analyze Arab sentiments, Arabic dialects, Ontology, Machine Learning.

Résumé

L'objectif de notre recherche est d'analyser les sentiments exprimés dans la langue arabe et ses dialectes. Cela implique d'extraire les émotions liées au voyage à partir des textes et de les analyser avec précision en fonction des différentes variantes dialectales arabes. Nous avons développé deux approches à cet effet. La première approche repose sur l'ontologie, tandis que la deuxième approche combine l'ontologie avec l'apprentissage automatique. Ces approches sont utilisées pour analyser des textes dans vingt-cinq dialectes différents.

Mots Clée : Analyse des sentiments, Analyse des sentiments Arabes, Dialectes Arabes, Ontologie, Apprentissage Automatique.

Contents

List of Figures

List of Tables

General Introduction

This project focuses on developing a multi-dialect Arabic sentiment analysis system to accurately determine sentiments expressed in different Arabic dialects. Arabic sentiment analysis presents unique challenges due to its morphological complexity and dialectal variations. Existing techniques often fail to handle these complexities effectively. The project aims to address these challenges by proposing a hybrid approach that combines lexicon-based methods, machine learning techniques, and ontology construction. The system leverages sentiment lexicons, develops a multi-dialect Arabic ontology, and applies advanced machine learning algorithms for improved accuracy and coverage. The thesis is organized into sections covering datasets, literature review, the proposed system, and experimental results to evaluate its performance. The project aims to contribute to Arabic sentiment analysis by developing a comprehensive system that can handle multiple dialects and improve sentiment analysis accuracy in Arabic text.

Chapter 1

Multi-dialect Arabic Sentiment Analysis

1 Introduction

Arabic sentiment analysis is the process of identifying and categorizing the sentiment expressed in text written in Arabic and different dialects of the same language as positive, negative, or neutral (Oxford Languages). These dialects include Egyptian Arabic, Levantine Arabic, Gulf Arabic, and Maghrebi Arabic. With the rise of social media platforms and the large volume of Arabic content available online, there has been an increasing need for Arabic sentiment analysis to understand people's opinions and attitudes towards different subjects. Additionally, it is a sub-field of natural language processing (NLP) and machine learning, which uses algorithms and models to classify the sentiment of text.

The importance of multi-dialect sentiment analysis lies in the fact that people express themselves differently across different dialects of the same language. Moreover, most of them use mixtures of many dialects in their speaking or writing. This is due to the cultural development of people and their familiarity with the dialects of others and their use in their daily lives. Therefore, it is essential to develop models that can accurately analyze sentiment across different dialects to gain a better understanding of public opinion in different regions.

Multi-dialect sentiment analysis can help individuals and organizations to better understand the emotions and attitudes of Arabic speakers towards different topics, products, or services. This can provide valuable insights into customer feedback, market trends, and social trends. For example, businesses can use Arabic sentiment analysis to understand the sentiments of Arabic-speaking customers towards their products and services and use this information to improve their offerings or address negative feedback.

Multi-dialect sentiment analysis is also important in social media monitoring, as it can help identify emerging trends and public opinions on specific topics. This can be useful for organizations and individuals to engage with their audience and understand their needs and preferences.

One of the major challenges of multi-dialect sentiment analysis is the lack of standardization in the language. Each dialect has its own unique characteristics, including different vocabulary, syntax, and morphology. Therefore, developing models that can accurately analyze sentiment across different dialects requires building large datasets that include text in different dialects and developing models that can handle the variations in the language.

Overall, Arabic sentiment analysis is a valuable tool for analyzing and understanding

the subjective information contained in Arabic text data. Its applications are diverse and wide-ranging, and it can provide valuable insights for individuals and organizations to make informed decisions and better understand the sentiments of Arabic-speaking audiences.

2 Problem Statement

Modern Standard Arabic (MSA) is very common all around the world in more than 25 countries. However, other Arabic dialects have appeared due to mixing pure MSA with different local languages. Basically, five famous categories of Arabic dialects are Levantine, Egyptian, Gulf, Iraqi, and Maghreb.

As a result, people in the Arabic world need to express their own dialect or sometimes a mixture of different dialects.

The main theme of this work is to analyze and evaluate the reviews of Arabic people in the travel domain written in multi-dialect Arabic. The reviews could convey positive sentiments, which mean satisfaction, or negative sentiments, which mean dissatisfaction.

Despite the extensive work done on sentiment analysis for MSA, a small number of research papers and reports are published on multi-dialect Arabic. This is certainly due to the difficulties associated with sentiment analysis from multi-dialect Arabic, which are explained in the next section.

3 Challenges Related to Arabic Sentiment Analysis

Arabic is widely spoken in many countries and regions, with different dialects, vocabulary differences, grammar differences, and cultural expressions. This makes sentiment analysis difficult. Therefore, we will highlight the challenges related to multi-dialect Arabic sentiment analysis:

Morphological complexity: Arabic has a complex morphological structure that involves variations in word forms [1], which makes it challenging to extract relevant features for sentiment analysis. Figure 1.1 shows an example of morphological complexity.

Dialectal Variation: Arabic spoken varies widely in terms of pronunciation, vocabulary, and grammar. It is spoken in many different dialects, each with its own unique features and vocabulary. This can make it difficult to build a robust sentiment analysis

	Meaning	Morpheme
وَسَيَفْعَلُهُ	And	وَ
	Future indication	سَ
	Present Tense,3rd,person,singular,masculine,inflection	يَ
	Root letters of verb (TO DO).	فْعَلُ
	Direct object pronoun,masculine,3rd person singular.	هُ

Figure 1.1: An example of Arabic morphological complexity.

system that can accurately analyze text from different dialects. Figure 1.2 provides a comparison of translations of a single sentence in different Arabic dialects.

English	*I Don't know what to i do*
Jordan	مش عارف شو اعمل
Palestinian	شو بدي اعمل
Emirati	معرف شو اسوي
Egyptian	مش عارف اعمل ايه
Tunisian	منعرفش
Algerian	ماعلباليش واش ندير
kuwaiti	ما ادري شو اسوي

Figure 1.2: A table comparing the translations of a single sentence in different Arabic dialects. English is the original sentence that has been translated into other Arabic dialects.

Lack of labeled data: There is a scarcity of labeled datasets for Arabic sentiment analysis, which makes it challenging to train and evaluate machine learning models.

Semantic ambiguity: Arabic words can have multiple meanings depending on the context in which they are used [2]. This can lead to ambiguity in sentiment analysis and affect the accuracy of sentiment prediction. Figure 1.3 shows an example of semantic ambiguity in the Arabic language.

Negation: Negation is an important aspect of sentiment analysis, but it can be challenging to identify negation in Arabic text due to the complex sentence structure and grammar. Figure 1.4 shows some Arabic negation words and inverters.

- قدّم = قدم = server
- قَدَم = قدم = foot

Figure 1.3: An example of semantic ambiguity in Arabic language.

MSA Inverters	DA Inverters	Translation
لا	مفي	does/do not
لم	مافي	does/do not
لما	منو	without
لات	ماكو	there is no
لن	مو	There will not be
بلا	بلاش	without
ليس	مش	No
من دون	مانو	without
بدون	م	without

Figure 1.4: Some Arabic negation words and inverters.

Sarcasm and irony: Arabic speakers often use sarcasm and irony in their communication, which can be difficult for sentiment analysis models to detect.

Data Collection: One of the major challenges in multi-dialect Arabic sentiment analysis is the scarcity of labeled data for each dialect. Collecting a sufficient amount of data for each dialect can be difficult, time-consuming, and expensive.

Code-Switching: Many Arabic speakers code-switch between dialects or mix dialects with Standard Arabic, which can create ambiguity and noise in sentiment analysis. This makes it difficult for sentiment analysis systems to accurately classify the sentiment of the text.

Domain Adaptation: Arabic sentiment analysis models trained on one domain may not perform well when applied to another domain due to differences in vocabulary and language used in each domain.

Lack of Standardization: There is no standard for writing Arabic text, which can create difficulties in preprocessing text and in building sentiment analysis models that can handle different writing styles and variations.

Sentiment Polarity: Arabic sentiment analysis faces the challenge of dealing with the high degree of subjectivity in Arabic language and culture, which can lead to different opinions and interpretations of the same text by different people. This can make it difficult to accurately determine the sentiment polarity of the text.

Limited Research: There is a limited amount of research on multi-dialect Arabic sentiment analysis, which can make it difficult to develop effective models and techniques for sentiment analysis in this context.

4 Novel Contributions of this Thesis

The proposed work introduces several novel contributions in the field of multi-dialect Arabic sentiment analysis:

- Building a large travel-domain ontology that consists of Modern Standard Arabic (MSA) terms and their synonyms in 25 Arabic dialects.
- Extracting sentiment polarities from multi-dialect Arabic reviews with high accuracy. The reviews can include up to 25 dialects.
- Developing a model that can handle negation and excessive words, improving the overall accuracy of sentiment analysis.
- Developing a translator system in the travel domain using a machine learning approach. The system can predict MSA synonyms for words that may belong to one of the 25 Arabic dialects.
- Proposing a statistical model to find the MSA synonym of any dialectal word that may be written incorrectly, as writing Arabic dialect words is more error-prone compared to MSA.

5 Thesis Organization

The proposed work is organized into several chapters, each focusing on different aspects of multi-dialect Arabic sentiment analysis. The organization of the thesis is as follows:

Chapter 1: - Provides an overview of the main theme of the proposed work, introduces the problem statement, discusses the challenges related to Arabic sentiment

analysis, and presents the novel contributions of the thesis.

Chapter 2: - Introduces the data sets used in the research, namely the Arabic Dialect Corpus and Lexicon (MADAR) and the Hotel Arabic Reviews (HARD) data set. It explains the characteristics of each data set and provides some samples.

Chapter 3: - Presents a comprehensive literature review of various approaches used in analyzing Arab sentiments. It discusses the existing research and methodologies in the field of Arabic sentiment analysis.

Chapter 4: - Clarifies in detail the proposed multi-dialect Arabic sentiment analysis system. It explains the methodology, algorithms, and techniques used in the system and provides a step-by-step explanation of the system's workflow.

Chapter 5: - Presents the most important findings of the research and analyzes the results obtained. It evaluates the performance of the proposed system and discusses the strengths and limitations of the approach.

Conclusion: - Summarizes the research and provides concluding remarks. It also discusses the future prospects and potential improvements for the proposed system.

The organization of the thesis ensures a comprehensive and structured exploration of multi-dialect Arabic sentiment analysis, starting from the introduction of the problem and challenges, followed by data set introduction, literature review, and detailed explanation of the proposed system. The results and analysis chapter provides insights into the system's performance, and the conclusion chapter summarizes the research and discusses future directions.

Chapter 2

Datasets

1 Introduction

This section introduces the two primary data sets used in the research: the Arabic Dialect Corpus and Lexicon (MADAR) and the Hotel Arabic Reviews (HARD) data set. It highlights the unique characteristics of each data set and provides sample excerpts to illustrate the nature of the data being analyzed.

2 The Arabic dialect corpus and lexicon(MADAR)

The Arabic Dialect Corpus and Lexicon (MADAR) is a linguistic resource developed by the Qatar Computing Research Institute (QCRI) to capture the linguistic diversity of Arabic dialects spoken across various regions of the Arab world [3]. The corpus contain more than 240,000 words in written texts and the lexicon contains 1042 concepts with average of 45 words per concepts in 25 different Arabic dialects from 13 different countries.

MADAR was compiled using a rigorous methodology that involved collecting spoken and written texts from various sources, including social media, news articles, and online forums. The texts were then transcribed and annotated with part-of-speech tags, lemmas, and morphological features.

The accompanying lexicon provides information about the following aspects of each word form:

Part of speech: Each word form is labeled with its part of speech, including nouns, verbs, adjectives, adverbs, prepositions, conjunctions, and interjections.

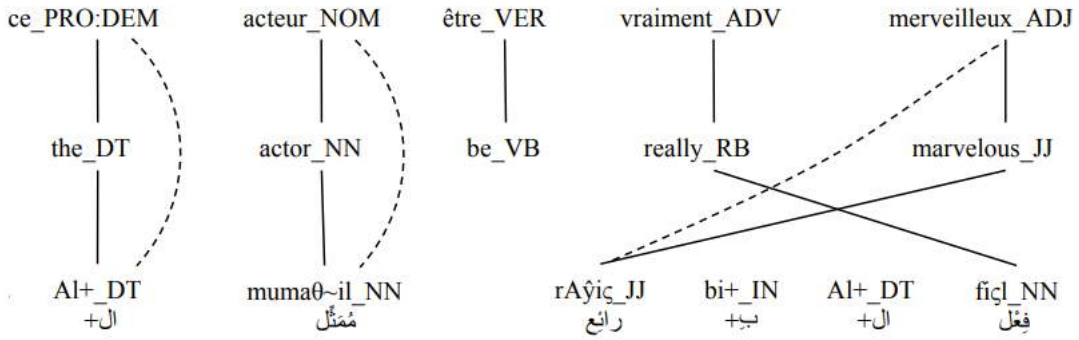


Figure 2.1: Clarifies Part of the "speech" description of the Arabic language.

Translation: Each word form is accompanied by its translation into Modern Standard Arabic, French and English.

English	French	MSA	Dialect	CODA
nice	gentil	لَطِيف	BEN	باهي
nice	gentil	لَطِيف	TRI	باهي
nice	gentil	لَطِيف	MUS	برابر
nice	gentil	لَطِيف	MUS	جميل
nice	gentil	لَطِيف	BEI	جنتيل
...	...	...	...	...
nice	gentil	لَطِيف	SAL	منيح
nice	gentil	لَطِيف	AMM	ناعم
nice	gentil	لَطِيف	JER	ناعم
nice	gentil	لَطِيف	SAL	ناعم
nice	gentil	لَطِيف	ALG	هايل

Figure 2.2: An example of translation in a MADAR lexicon.

Figure 2.2 introduces an example of MADAR lexicon where the first column presents the word in English, the second column presents it in French, the third column shows the word in MSA,the CODA column contains the words translated from MSA to the 25 dialects and the Dialect column specifies the city (dialect location).

Dialect: The lexicon provides information about the dialect(s) in which each word form is used, including the dialectal variations in pronunciation, morphology, and syntax.

Morocco	Algeria	Tunisia	Libya	Egypt/Sudan	South Levant	North Levant	Iraq	Gulf	Yemen
Rabat (*RAB*) Fes (*FES*)	Algiers (*ALG*)	**Tunis** (*TUN*) Sfax (*SFX*)	Tripoli (*TRI*) Benghazi (*BEN*)	**Cairo** (*CAI*) Alexandria (*ALX*) Aswan (*ASW*) Khartoum (*KHA*)	Jerusalem (*JER*) Amman (*AMM*) Salt (*SAL*)	**Beirut** (*BEI*) Damascus (*DAM*) Aleppo (*ALE*)	Mosul (*MOS*) Baghdad (*BAG*) Basra (*BAS*)	**Doha** (*DOH*) Muscat (*MUS*) Riyadh (*RIY*) Jeddah (*JED*)	Sana'a (*SAN*)

Figure 2.3: Different City dialects in the MADAR resources.

Frequency: The lexicon includes information about the frequency of each word form in the corpus, as well as its relative frequency across the different dialects.

Morphological features: The lexicon provides information about the morphological features of each word form [3], including its root, stem, tense, aspect, mood, and gender.

The MADAR Arabic Dialect Corpus and Lexicon has many applications in natural language processing, machine learning, and linguistic research. It can be used to train

language models, develop computational tools to process Arabic dialects, and study the phonological, morphological, and syntactic features of Arabic dialects. The corpus and lexicon are freely available for research purposes.

In addition to the MADAR Corpus, the QCRI has also developed a range of other resources for Arabic dialects, including the MADAR Lexicon and the MADAR POS Tagset [4]. These resources provide detailed information about the vocabulary, grammar, and syntax of Arabic dialects, and can be used to develop more accurate and effective NLP applications for these dialects.
Overall, The MADAR Arabic Dialect Corpus and Lexicon is an important resource for researchers and developers working on natural language processing (NLP) applications for Arabic dialects. It is particularly valuable because it includes data from a wide range of dialects, which vary significantly in their vocabulary, grammar, and pronunciation. This makes it possible to train and test NLP systems that can handle the diversity and complexity of Arabic dialects.

3 HARD: hotel Arabic reviews dataset

HARD (Hotel Arabic Reviews Data set) is a large data set of hotel reviews written in Arabic language [5]. It was created to support research in natural language processing (NLP) and sentiment analysis in Arabic language, specifically in the context of hotel reviews.

The dataset contains 93700 hotel reviews written by both native and non-native Arabic speakers. Divided by a balanced and unbalanced reviews. Each review is associated with a rating on a 5-point scale, Positive (rating 4 and 5), negative(ratings 1 and 2). In addition to the reviews themselves, the dataset includes metadata such as the hotel name, location, and date of the review.

Property	Number	Property	Number
Number of reviews	373,750	Median reviews per hotel	150
Number of hotels	1,858	Min reviews per hotel	3
Avg. reviews per hotel	264	Number of users	30,889
Max reviews per hotel	5,793	Avg. reviews per user	15.8
Median reviews per hotel	150	Number of tokens	8,520,886

Figure 2.4: Some properties of HARD dataset.

The balanced subset of the HARD dataset refers to a subset where the number of positive reviews (ratings 4 and 5) and negative reviews (ratings 1 and 2) are evenly distributed. In other words, it contains an equal number of reviews with positive and negative sentiments.

On the other hand, the unbalanced subset of the HARD dataset refers to a subset where the distribution of positive and negative reviews is not equal. This means there may be a significant difference in the number of positive and negative reviews.

Both the balanced and unbalanced subsets serve different purposes and can be used for various research objectives. The balanced subset provides a fair representation of positive and negative sentiments, while the unbalanced subset allows for analyzing scenarios with imbalanced data.

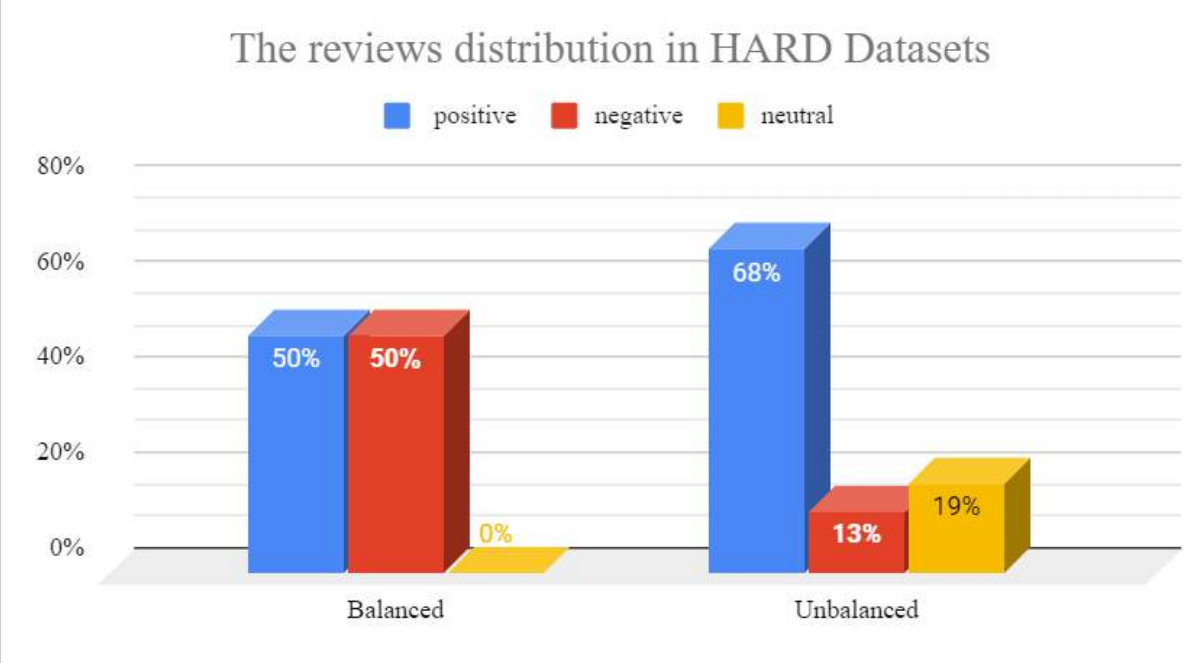


Figure 2.5: The reviews distribution in Hard dataset.

For our study, we have opted to work with the balanced subset of the HARD dataset, which consists of an equal number of positive and negative hotel reviews. This decision has been made to ensure a fair representation of both positive and negative sentiments in our analysis. By maintaining this balance, we can avoid biases and gain a comprehensive understanding of sentiment patterns in Arabic hotel reviews.

The use of balanced reviews also allows for reliable evaluation, as metrics like accuracy can provide a meaningful measure of effectiveness across different sentiment classes. Additionally, the balanced subset enables comparative analysis, facilitating fair comparisons between different techniques or approaches. Overall, the inclusion of balanced reviews enhances the integrity and depth of our study, allowing us to explore sentiment analysis in the context of Arabic hotels more effectively.

HARD has been used in a variety of research studies and experiments, including sentiment analysis, text classification, machine translation, and more. Researchers and developers have used the dataset to train and evaluate various machine learning models.

We have selected the HARD dataset due to its inclusion of Arabic reviews, encompassing both Modern Standard Arabic (MSA) and various Arabic dialects. Although it primarily focused on the domain of hotels, the dataset holds relevance for our research, as it includes terminology that aligns with the lexicon found in the MADAR dictionary. By utilizing the HARD dataset, we can effectively analyze sentiment in Arabic and its dialects, while drawing connections to the comprehensive linguistic resource provided by

MADAR.

4 Conclusion

In conclusion, this chapter has presented an overview of the data sets utilized in the research: the Arabic Dialect Corpus and Lexicon (MADAR) and the Hotel Arabic Reviews (HARD) data set. The distinct characteristics of each data set have been discussed, and sample excerpts have been provided to demonstrate the nature of the data. These data sets will serve as valuable resources for conducting sentiment analysis on Arabic text, facilitating further investigation and analysis in this field.

Chapter 3

Literature Review

1 Introduction

In this chapter, we delve into the literature reviews on various approaches to Arabic sentiment analysis. We explore the existing methods and techniques employed to analyze sentiment in Arabic text, aiming to gain insights into the advancements made in this field.

2 Lexicon-based approach

This approach uses a lexicon or dictionary of words and their corresponding sentiment scores to analyze the sentiment in the text [6]. Sentiment scores for words in the text are aggregated to calculate an overall sentiment score for the text. The lexicon can be created manually or generated automatically.

In these reviewed studies which are shown below in Table [3.1] , the lexical approach was used, which involves calculating the degree of polarity for each word in the commentary. There are four studies on the three dialects: Egyptian, Gulf, and Algerian. Encouraging accuracy of 98.20% and 93.20% was achieved by [7] on a dataset collected on Facebook from 3484 Egyptian comments. Another study had acceptable results of 77.4% (positive), 59.1% (negative), and 51.1% (neutral) on a data set collected from 1,500 Gulf tweets [8] with a percentage of 85.40% in the data. 7,698 Gulf tweets were collected. [9]. Whereas [8] achieved an accuracy of 79.13% on a dataset collected on Facebook of 7,698 Algerian comments.

Reference	Dataset	Dialect	Labels	Performance
[7]	3484-Facebook	Egyptain	Pos	98.2-% Acc
			Neg	93.20% Acc
[9]	4700-Twitter	Gulf	Pos, Neg	85.40% Acc
[8]	1500 - Twitter	Gulf	Neg	77.4% F1-score
			Pos	59.1% F1-score
			Neu	51.1%F1-score
[8]	7698 - Facebook	Algerian	Pos, Neg, Neu	79.13% Acc

Table 3.1: Sample of previous studies that belong to the lexicon-based approach.

3 Machine learning approach

This approach uses machine learning algorithms to learn patterns in the data and predict the sentiment of a text. The algorithm is trained on a labeled data set [10], where each text is labeled with its corresponding sentiment. Once trained, the algorithm can be

used to predict the sentiment of new texts .

Studies based on machine learning approaches have recorded high accuracy rates, some of which we summarized in terms of accuracy, algorithm, dataset, dialect, and labels. As shown in Table [3.2] below. The highest accuracy of 98.04% was recorded when combining RF classification and TF-IDF techniques with a data set of 5,615,943 by [11].

Ref	Dialect	Algorithm	Feature extractor	Dataset	Labels	Accuracy
[12]	Levantine	SVM	BOW	1300 - Facebook	Pos, Neg, Neu	97.90%
[11]	Multi-dialects	LR	TF-IDF	5615943 - Twitter	Pos, Neg, Neu	97.72%
[13]	Multi-dialects	AdaBoost	AraBert	91000 - Website	Pos, Neg	84.20%
[14]	Maghrebian	ME	Word2vec	3355 - Multi-dialects	Pos, Neg	83.9%
[12]	Levantine	KNN	BOW	1300 - Facebook	Pos, Neg, Neu	96.80%
[15]	Multi-dialects	SGD	Word2vec	1951 - Twitter Website	Pos, Neg	79.52%
[16]	Maghrebian	SGD	Word2vec	254000 - Facebook	Pos, Neg	90.16%
[11]	Multi-dialects	DT	TF-IDF	5615943 - Twitter	Pos, Neg, Neu	97.71%
[11]	Multi-dialects	RF	TF-IDF	5615943 - Twitter	Pos, Neg, Neu	98.04%

Table 3.2: Sample of previous studies that belong to the machine learning Approach.

As shown in Table [3.3], deep learning has been shown to give highly accurate ratios. The best performances were scored with an accuracy of 99.82%, with the CNN technique combined with Word2vec in [17] on the dataset (5615943 - Twitter). As well as using DL architectures comprising RNN Types such as GRU and LSTM, gave high results. For example the study conducted by [14] using LSTM achieved an accuracy of 97.20%.

Ref	Algorithm	Feature extractor	Dataset	Dialect	Labels	Acc
[11]	mLSTM	AraVec	5615943 - Twitter	Multi-dialects	Pos, Neg Neu	99.75%
[17]	CNN	Word2vec	5600 - Twitter	Multi-dialects	Anger, Fear Joy,Sadness	99.82%
[14]	99f LSTM	Word2vec	2000 - Multi	Maghrebian	Pos, Neg	97.20%
[13]	CNN-BiLSTM	AraBert	91000 - Website	Multi-dialects	Pos, Neg	94.2%
[13]	CNN-LSTM	AraBert	91000 - Website	Multi-dialects	Pos, Neg	94%
[11]	GRU	AraBert	5615943 - Twitter	Multi-dialects	Pos, Neg Neu	98.80%
[18]	BiGRU	AraBert	56674 - Twitter	Gulf	Pos, Neg Neu	81.59%

Table 3.3: Sample of previous studies that belong to the deep learning Approach.

4 Hybrid approach

This approach combines the lexicon-based approach and the machine learning approach to improve the accuracy of sentiment analysis [19]. The lexicon is used to provide an initial sentiment score, which is then refined using machine learning algorithms. This approach can be particularly effective in cases where the lexicon-based approach may not work well, such as with slang or new words that are not in the lexicon.

In most studies where machine learning was combined with a lexicon-based approach, the SVM classifier was often used, which gave very good results. An accuracy of 95.70% was reached on a dataset of 2000 comments collected from Twitter and websites in Egypt. In other studies using the same classifier and different databases, accuracy ranged from 95.6% to 60.60%. The latter was achieved using a multi-dialect datasets drawn from 1,200 tweets. In Table [3.4] below is a sample of studies that relied primarily on the hybrid approach.

Ref	Dataset	Dqilect	Labels	Classifier	Accuracy
[16]	2000 - Websites Twitter	Egyptian	Pos, Neg, Neu	SVM	95.70%
[20]	2730 - Website	Jordanian	Pos, Neg	SVM	95.60%
[16]	7366 - Twitter Facebook	Tunisian	Pos, Neg	SVM	94%
[21]	2500 - Website	Jordanian	Pos, Neg	KNN	91%
[22]	943 - Twitter	Multi-dialects	Pos, Neg	SVM	91%
[23]	1111 – Twitter	Egyptian	Pos, Neg	Bagging with SVM	90%
[24]	1200 - Twitter	Multi-dialects	VeryPos, Pos, Neu, Neg, VeryNeg	SVM	60.60%

Table 3.4: Sample of previous studies that belong to the Hybrid Approach.

The key difference between our approach and other approaches lies in the integration of machine learning, statistics, and ontology. By combining these techniques, we aim to address the challenges posed by multi-dialect reviews and improve the accuracy and robustness of sentiment analysis and translation tasks. Our approach takes into account dialect-specific variations, leverages domain-specific knowledge through ontology, and incorporates statistical models for handling negation and errors in dialectal word writing.

5 Conclusion

To conclude this chapter, we have observed that Arabic sentiment analysis approaches can be broadly categorized into three main types: hybrid approaches, machine learning-based approaches, and lexicon-based approaches. These approaches offer diverse methodologies to effectively analyze sentiment in Arabic text, providing a foundation for further research and development in this area.

Chapter 4

The Proposed Multi-dialect Arabic Sentiment Analysis System

1 Introduction

In this chapter we will take some proposed methods that we've designed to extract the sentiment from a given sentence, essentially there are two main models proposed **Stemming-based sentiment analysis system** and **Machine learning-based sentiment analysis system**, these two models have the same based architecture which means they are both an **Ontology-based models**, also we will be discussing some shared and special data processing for each model, as it's known the data processing and building these models having also shared and special tools and techniques.

2 Working environment

2.1 Hardware environment

Laptop: I5 11 th Gen, RAM: 8 GB.
Laptop:I3 3 th Gen, RAM: 6 GB.
PC: I3 10 th Gen, RAM: 16 GB, GPUs GTX 1060.
PC: Ryzen 3 3200g, RAM: 16 GB.

2.2 Softwar environment and libraries

- **Protege(5.5.0):** Protege is an open-source software platform used for building and managing ontologies [25]. It provides a user-friendly interface and supports various ontology languages like RDF(S) and OWL, with Protege, users can define classes, properties, relationships, and constraints within an ontology. It also offers advanced features like reasoners for automatic inference and validation.
- **Visual Studio Code(1.73.1):** Commonly known as VS Code, is a lightweight yet robust source code editor that can be installed on your desktop. It is compatible with Windows [26], mac OS, and Linux operating systems. While it offers built-in support for popular programming languages like JavaScript, TypeScript, and Node.Js, it also boasts a wide range of extensions to cater to other languages and run times such as C++, C#, Java, Python, PHP, Go, and .NET.Version.
- **Python(3.9.12):** Python is a versatile and beginner-friendly programming language widely used for various applications [27], including web development, data analysis, scientific computing, and automation.
- **TensorFlow(2.10.0):** TensorFlow is a popular open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools [28], libraries, and resources for building and deploying machine learning models.

- **Sklearn(1.2.1):** Scikit-learn (sklearn) is a comprehensive machine learning library for Python [29], providing a wide range of algorithms and tools for various machine learning tasks, such as classification, regression, clustering, and dimensionality reduction.

- **NumPy(1.24.1):** NumPy is a Python library for numerical computing that enables efficient and high-performance operations on multidimensional arrays [30], essential for tasks such as scientific computing, data analysis, and machine learning.

- **Pandas(1.5.3):** Pandas is a widely-used open-source library for data manipulation and analysis in Python [31]. It offers a user-friendly and efficient way to handle structured data using its DataFrame data structure. With Pandas, you can easily clean, filter, transform, and aggregate data. It also provides robust tools for data visualization and seamless integration with other libraries. Pandas is a popular choice among data scientists and analysts for working with tabular data in Python.

- **Tashaphyne(0.3.2):** Tashaphyne is an Arabic light stemmer and segmenter that focuses on removing prefixes and suffixes from Arabic words while generating all possible segmentations [32]. It offers stemming and root.

- **ISRIStemmer(3.7):** The ISRI stemmer is part of NLTK, a widely used Python library for natural language processing [33]. By importing the nltk.stem.isri module, you can access the ISRI stemmer and leverage its stemming capabilities in your Python code. This integration enables you to utilize the ISRI stemmer alongside other useful functionalities provided by NLTK for Arabic text processing.

- **CamelTools(1.4.1):** CamelTools is an NLP toolkit specifically designed for Arabic text processing [34]. It offers a range of linguistic resources and algorithms to perform tasks like tokenization, stemming, part-of-speech tagging, and named entity recognition. CamelTools enhances Arabic text analysis and is widely utilized in research and industry applications.

- **Owlready2(0.39)** Owlready2 is a Python library for working with OWL ontologies [35]. It provides an intuitive interface for loading, creating, and manipulating ontologies, allowing developers to perform various operations and reasoning tasks. It is widely used in ontology engineering and semantic web applications.

- **word2vec:** It is a family of model architectures and optimizations that can be used to learn word embeddings from large datasets [36]. Embeddings learned through word2vec have proven to be successful on a variety of downstream natural language processing tasks.

- **Recurrent Neural Networks (RNN):** Traditional neural networks have limitations when it comes to processing sequential data, such as understanding the context of events in a movie [37]. Recurrent neural networks (RNNs) address this issue by incorporating loops in their structure, allowing information to persist and be used to inform later steps.

RNNs can be visualized as multiple copies of the same network, with each copy passing information to the next one. This chain-like structure reveals the close relationship between RNNs and sequences or lists of data.

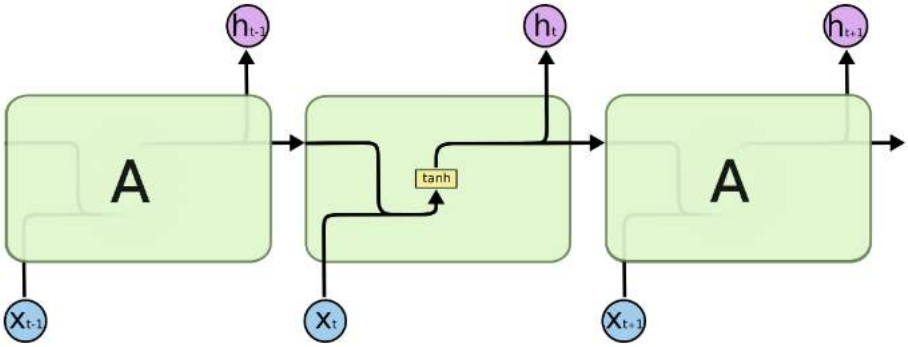


Figure 4.1: The repeating module in a standard RNN contains a single layer.

RNNs, particularly a specialized type called Long Short-Term Memory (LSTM), have shown remarkable success in various applications, including speech recognition, language modeling, translation, and image captioning. LSTMs have significantly outperformed the standard RNNs in many tasks [38], and they are the focus of exploration in this essay.

- **LSTM Networks:** LSTMs, which stands for Long Short-Term Memory networks, are a specialized type of recurrent neural networks (RNNs) that excel at capturing long-term dependencies in sequential data. They were initially introduced by [39] and have since gained popularity and been refined by numerous researchers.

One of the main advantages of LSTMs is their ability to overcome the long-term dependency problem, which refers to the difficulty of traditional RNNs in retaining information over extended periods. LSTMs are specifically designed to address this issue and have a default behavior of effectively remembering information for longer durations.

Like other RNNs, LSTMs have a chain-like structure with repeating modules. However, the structure of the repeating module in LSTMs differs from that of standard RNNs. While a standard RNN module may consist of a single layer [40], LSTMs have four interconnected layers that interact in a unique way.

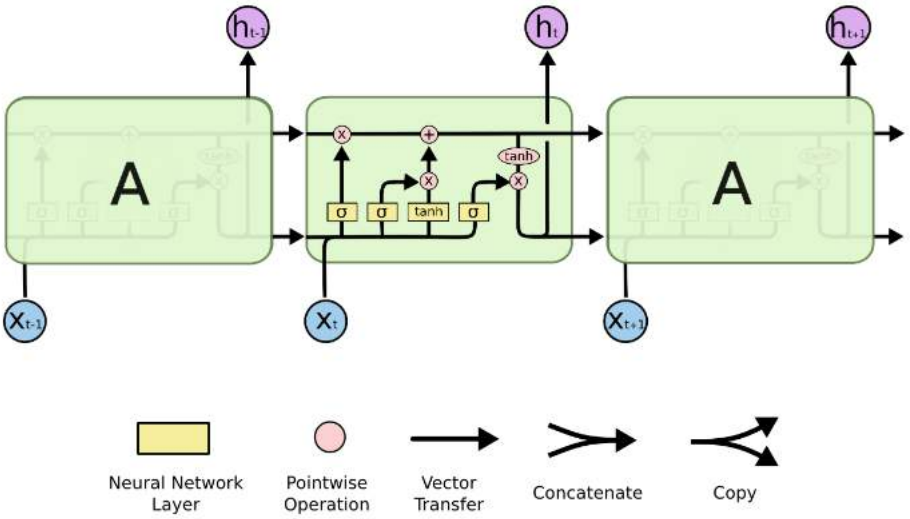


Figure 4.2: The repeating module in an LSTM contains four interacting layers.

The distinct architecture of LSTMs contributes to their superior performance across a wide range of problems, making them a widely adopted solution in the field of deep learning.

3 Data preprocessing

In this section, we will discuss the essential general pre-processing steps applicable to any sentiment analysis task. These steps include text cleaning, tokenization, stop-word removal, and handling special characters or symbols. By following these best practices, we can ensure that the data is appropriately pre-processed for subsequent analysis.

Specific pre-processing steps tailored to the individual models proposed for our case study will be discussed in their respective model sections. This approach allows us to address any unique requirements or characteristics of each model individually. By considering both general and specific pre-processing steps, we aim to enhance the overall performance and accuracy of the sentiment analysis models.

3.1 MADAR Lexicon dataset

First the dataset wasn't in the right shape for our proposed models to handle, therefore starting with:

- Removing the **useless columns** from the dataset.
- Removing rows that have white space, special character and sentence:
- Splitting the dataset into three main sections neutral, negative and positive, each word has been labeled according to its polarity **"1" as positive, "0" as negative- and "2" as neutral**, this step shared between three different people to labeling the words by majority vote.

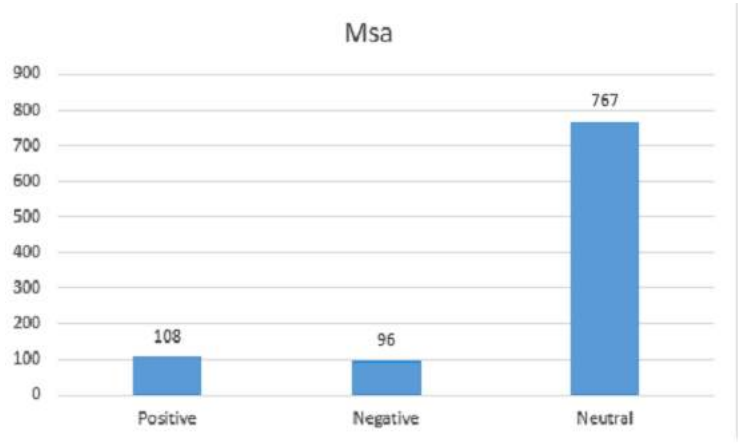


Figure 4.3: Sentiment distribution in MSA.

Figure 4.3 provides an overview of the sentiment distribution specifically within the MSA words of the MADAR lexicon dataset. It helps in understanding the distribution of positive, negative, and neutral sentiments in the MSA words and can be useful for sentiment analysis tasks and understanding the general sentiment tendencies within the available Modern Standard Arabic in the MADAR lexicon dataset.

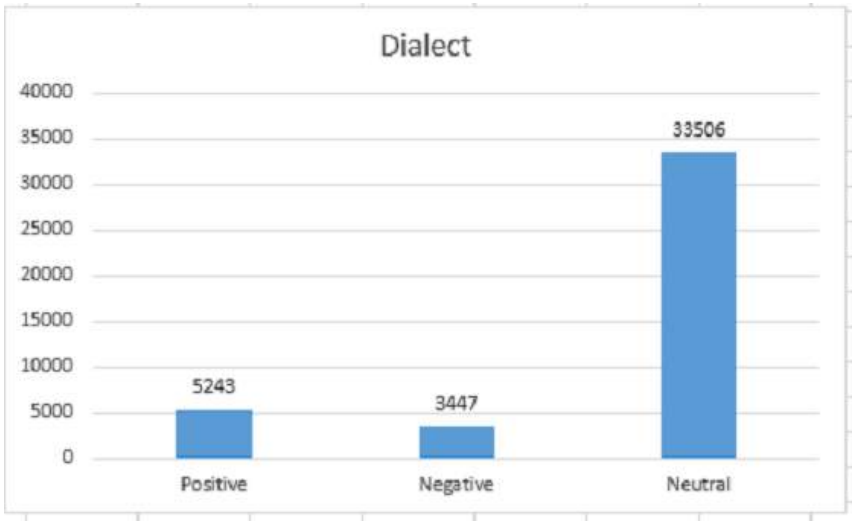


Figure 4.4: Sentiment distribution in Dialect.

Figure 4.4 provides insights into the sentiment distribution specifically within the Arabic dialect words of the MADAR lexicon dataset. It helps in understanding the distribution of positive, negative, and neutral sentiments in the dialectal variations and can be valuable for sentiment analysis tasks specific to Arabic dialects.

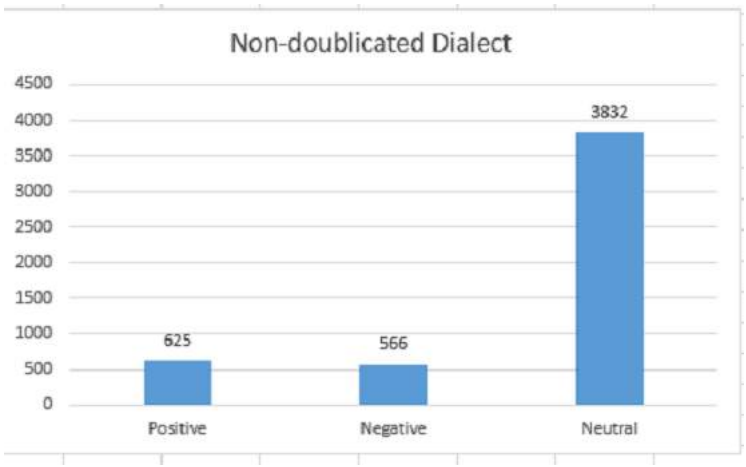


Figure 4.5: Sentiment Distribution in Non-Duplicated Dialect.

Figure 4.5 represents the sentiment distribution within a modified dataset created by removing duplicate entries from the Dialect column of the MADAR lexicon dataset. This modified dataset focuses on unique dialect words, providing a clearer view of sentiment distribution while minimizing data redundancy.

By examining the sentiment distribution in the Non-Duplicated Dialect dataset, we can gain insights into the sentiment landscape within unique Arabic dialect words. This refined view helps in understanding the prevalence and distribution of positive, negative, and neutral sentiments within the distinct dialect variations present in the dataset.

Together, these figures contribute to a comprehensive understanding of sentiment distribution within the MADAR lexicon dataset, covering sentiment patterns in both Modern Standard Arabic and Arabic dialects. They highlight the unique characteristics of each language variant and facilitate sentiment analysis tasks specific to different forms of Arabic, aiding us in exploring sentiment nuances and variations within the language.

3.2 HARD Hotel Arabic Reviews Dataset

We took 1000 of (HARD Hotel Arabic Reviews Dataset) randomly selected, the first half represents positive reviews and the other half represents negative reviews. The content and nature of the reviews may vary, as they represent real-world sentiments expressed by individuals regarding the specific subject of our analysis (e.g., hotel experiences in the case of HARD Hotel Arabic Review). The reviews might contain opinions, feedback, or

subjective evaluations regarding different aspects of the subject, such as service quality, facilities, location, or overall experience.

The content and nature of the reviews may vary, as they represent real-world sentiments expressed by individuals regarding the specific subject of our analysis (e.g., hotel experiences in the case of HARD Hotel Review). The reviews might contain opinions, feedback, or subjective evaluations regarding different aspects of the subject, such as service quality, facilities, location, or overall experience.

By manually labeling the reviews as positive or negative, we have created a ground truth for sentiment analysis, which serves as a basis for training and evaluating sentiment classification models. This labeling allows the models to learn patterns and characteristics associated with positive and negative sentiments, enabling them to classify new, unlabeled reviews in the future accurately.

It is important to note Neutral reviews typically refer to reviews that do not express strong positive or negative sentiment. They may contain factual information or opinions that are neither strongly positive nor negative. In the context of sentiment analysis, identifying and categorizing neutral reviews can be challenging as they lack clear sentiment polarity.

In our work, we need to add some neutral reviews to our collected hotel review dataset. These neutral sentences have been extracted from The MADAR Arabic Dialect Corpus Dataset (250 Sentence), taking ten sentences for each one of the twenty-five Arabic dialects. It's important to note that these sentences are not actual reviews, but rather facts and questions related to the lexical fields of traveling..

As every existing data we have starting with processing and reshaping the data to fit in to the models, beginning with:

- Removing special characters from the sentences in (1000 HARD Hotel Arabic Reviews Dataset + 250 The MADAR Arabic Dialect Corpus Dataset).
- Feeding the 1000 review with dialect words that extracted from The MADAR Lexicon, we have done this step because the dataset was filled with MSA words and this is not the goal of our project, this process was completely manually, it's based on reading the review and try to find for each word of it, what corresponds to it in our MADAR lexicon, if it's found replace it with the corresponds dialect word, and if wasn't found leave it, to facilitate this process we split the data into 25 section**(representing the 25 Cities that's The MADAR lexicon contains)**.

Each section contains 40 reviews the half of it is positive and the other half is negative to keep the balance.

Cities	Modified	Original
Aleppo	الموقع بجنن وحلو وسعرو مناسب , خدمة العفش بطيئة	ممتاز وحلو وسعرو مناسب. الموقع , النظافة. خدمة العفش بطيئة
Algiers	رائع الهندية في الاستقبال المسائي هايله جونتي بزاف وثانيك الفلبينيه والاستاذ المصري	ممتاز. الهندية في الاستقبال المسائي مميزه ولطيفه جدا وكذلك الفلبينيه والاستاذ المصري
Alexandria	ضعيف. مستوى النظافه والخدمات الفندقيه وحشه كثير	ضعيف. مستوى النظافه والخدمات الفندقيه سيئه للغايه
Amman	منيح كثير كويس و رايق الانترنت سريع	جيد جداً. جيد وهادي. الانترنت سريع
Aswan	ما يستاهل الخمسة نجوم. لقربه من الحرم. مستوى الخدمه ضعيف و وسخ كثير وانترنت ضعيف و السرفيس معفن	لا يستحق الخمسة نجوم. قربه من الحرم. مستوى الخدمه والنظافة جدا متدني وانترنت ضعيف و سوء الخدمه
Baghdad	زين هوايا. و قريب من المسجد الحرام	جيد جداً. القرب من المسجد الحرام.
Basra	مو زين ابدا . ريحة مكان البوفية خايسه و الاكل سئي جدا	مخيب للأمل . ريحته مكان البوفية خايسه و الاكل سئي جدا
Beirut	حلو كثير . الموقع بيعقد و قريب من الحرم.	جيد جداً. الموقع ممتاز قريب من الحرم.
Benghazi	ما بنصحش بيه. الفطور بس. موقع الفندق عالي والوصول له يتعب كتير والسعر ما يتناسبش مع مستوى الغرفة بالنظافة والخدمات وضيق الغرفة.	لا انصح به. الفطور فقط. موقع الفندق مرتفع والذهاب إليه متعب جدا والسعر لا يناسب مستوى الغرفه من ناحية النظافة والخدمات وضيق الغرفة
Cairo	الفطور حلو كثر السحر كويس الهدوء جميل مواقف السيارات متوفره	جيد. الفطورالسحرالهدوء. مواقف السيارات
Damascus	منو نظيف والحمامات تحتاج تصليح	جيد. عدم النظافه. والحمامات تحتاج تصليح ونظافه
Doha	زيبيين واجد الموقع حلو و نظيف	جيد جداً. موقعه و النظافه.
Fes	هايل بزاف و السعر مناسب و موقع الفندق زوين	جيد. الفندق طيب وسعره مناسب.
Jeddah	انصح فيه كويس كثير كل شي فيه رخيص سعره مقارنة بمستواه	رخيص سعره مقارنة بمستواه. انصح فيه جدا ممتاز كل شي فيه.
Jerusalem	رحلة حلوة المكان جميل و الاوتيل مريح.	رحلة جميلة. الموقع ممتاز والفندق مريح.
Khartoum	فندق زفت وما مريح. الفندق زقيل ما عندو صيانة للحمامات والتعاون مع العمال سيئ كتير.	فندق قديم وغير مريح. لا شي. الفندق قديم لا توجد صيانه للحمامات وتعامل الموظفين سيء جدا
Mosul	تووووب كل شيء اعجبني الغرفة والخدمات وحوض السباحة نظيف و واسع	ممتاااز. ممتاااز كل شيء اعجبني الغرفة والخدمات وحوض السباحة والسبا.
Muscat	كويس مره . حبذا لو يتم تغيير السجاد الأرضي لأنه اصبح قديم و غير نظيف	جيد. . حبذا لو يتم تغيير السجاد الأرضي لأنه اصبح قديم و غير نظيف
Rabat	هاايل عجبني ف كل شي بصراحه الفندق زوين بزاف وكذلك الخدمات. سعر غالي شويا	ممتاز. الفندق جميل جدا وكذلك الخدمات. سعر غالي جدا
Riyadh	الفندق يجنن و مريح نفسيا والقائمين عليه مقابلتهم حسنة و لطيفه كثير	فندق ممتاز. الفندق ممتاز ومريح نفسيا والقائمين عليه مقابلتهم حسنة.
As-Salt	غير مرضي. مكان الغرفه سيء .و نصب في حجز الغرف تعامل العمال زفت	غير مرضي. مكان الغرفه وتعامل الموظفين. خداع في حجز الغرف
Sana'a	كل شيء زفت. الاثاث قديم ويفتقد للنظافة والخدمة سيئة لأبعد حد	ضعيف. الموقع فقط. الاثاث قديم موظفو الاستقبال الفندق يحتاج الى ترميم كامل

Figure 4.6: Samples of original Vs modified Hard Reviews.

- Merge the (1000 HARD Hotel Arabic Reviews Dataset + 250 The MADAR Arabic Dialect Corpus Dataset) in one dataset.
- Tokenizing each sentence to a list of words using **Camel tool**.
- Remove the stuttering from the words of each review.

4 Stemming-based Sentiment Analysis System

Stemming is the process of reducing a word to its base or root form, also known as the stem, by removing any suffixes or prefixes that may be attached to it. Stemming is commonly used in natural language processing tasks [41], such as information retrieval, text classification, and sentiment analysis, to improve the accuracy of text analysis.
Here's an example of stemming in Arabic:

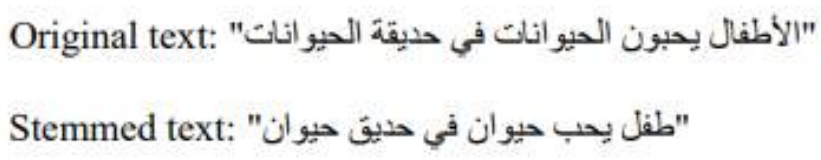


Figure 4.7: An example of stemming in Arabic.

In this example, the original text contains several Arabic words . After applying stemming, the words have been reduced to their base form or stem. This can help in identifying the main topics or themes present in the text, and can be useful in text classification or information retrieval tasks.

There are different approaches to stemming, ranging from simple rule-based algorithms to more complex statistical and machine learning-based methods. The choice of stemming algorithm depends on the specific use case and the characteristics of the text being processed. In the Figure 4.8 below our proposed model.

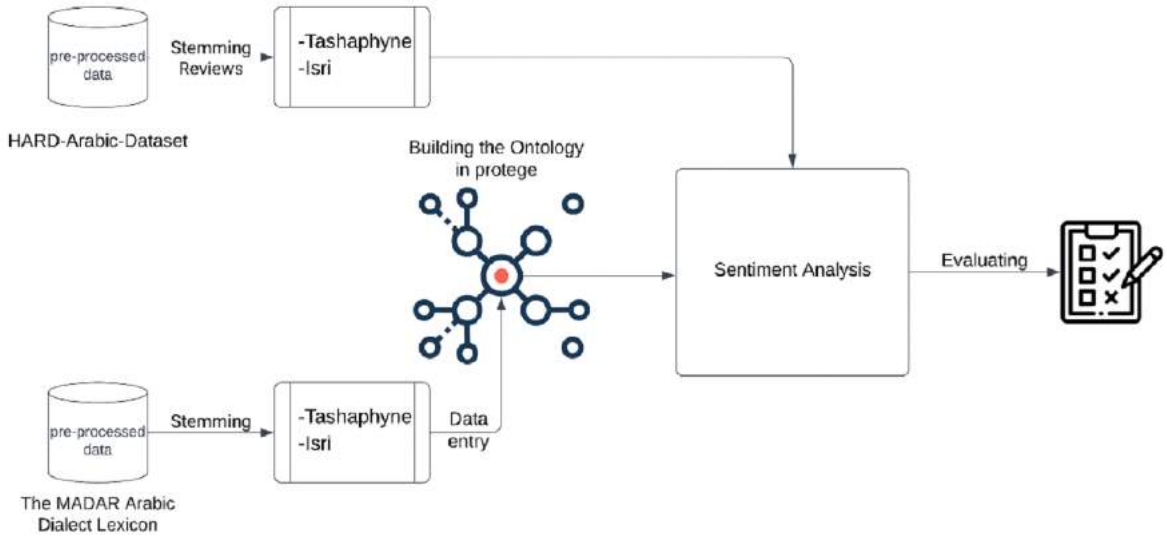


Figure 4.8: Stemming-based Sentiment Analysis System Architecture.

4.1 Specific Datasets Preprocessing

In this phase we do stem every word of our pre-processed MADAR lexicon with (Tashaphyne, Isri) tools, as shown in Figure 4.9.

	MSA	Dialect	CODA	TashaphyneStem	IsriStem
0	رَقْم ، عَدَد	AMM	رقم	رقم	رقم
32	رَقْم ، عَدَد	ALX	عدد	عدد	عدد
47	رَقْم ، عَدَد	RAB	نمره	مر	نمر
58	رَقْم ، عَدَد	ALG	نميرو	ميرو	يرو
59	رَقْم ، عَدَد	SFA	نومرو	ومرو	مرو
61	فُنْدُق	FES	اطل	طل	اطل
63	فُنْدُق	AMM	اوتيل	وتيل	ويل
68	فُنْدُق	ALG	اوتال	وتال	وال
75	فُنْدُق	FES	فندق	دق	ندق
104	فُنْدُق	BAG	فندقه	دق	ندق
107	فُنْدُق	KHA	لكونده	كوند	كوند
108	فُنْدُق	ALX	لوكنده	وكند	كند
112	فُنْدُق	KHA	هوتيل	هوتيل	هوتيل
116	فُنْدُق	SFA	وتيل	يل	وتل
118	فُنْدُق	FES	وطيل	طيل	وطل

Figure 4.9: Stemmed dialect words on MADAR dataset using (Tashaphyne, Isri).

Also Stem the pre-processed Hard Hotel review dataset to get it ready for sentiment analysis.

```
sentenceTokens    [كانت، اقامتي، سيئه، كتير، لا، خدمات، ولا، سرفس]
TashaphyneStem              [انت، اقام، ئه، تير، لا، خدم، لا، سرفس]
IsriStem                    [كانت، قمت، سيئ، كتر، لا، خدم، ولا، رفس]
Name: 20, dtype: object
```

Figure 4.10: Stemmed review in HARD dataset using (Tashaphyne, Isri).

4.2 Multi-dialects Arabic Ontology construction using stems

In sentiment analysis, ontology-based approaches offer a structured and flexible framework for extracting and classifying sentiment from text. When dealing with multi-dialects Arabic sentiment analysis, constructing an ontology becomes crucial in capturing the nuances and variations of sentiment expressions across different dialects. This section explores the key components of ontology construction, namely terms/concepts and relationships, in the context of multi-dialects Arabic sentiment analysis.

Terms/Concepts: The construction of a multi-dialects Arabic sentiment ontology involves identifying and defining sentiment-related terms and concepts. These terms encompass words or phrases that express sentiment or emotion in various Arabic dialects. The ontology should encompass a wide range of dialect-specific sentiment vocabulary, including positive and negative sentiment words, emotion-related terms, and dialect-specific expressions conveying sentiment. Each term should be appropriately labeled, defined, and described using attributes such as synonyms or variations in each dialect.

Relationships: In the realm of multi-dialects Arabic sentiment analysis, relationships within the ontology play a vital role in capturing the connections and associations between sentiment-related terms. These relationships provide a contextual framework for organizing sentiment knowledge across different dialects. They can reflect the similarities and differences in sentiment expression, contextual dependencies, and dialect-specific sentiment variations. For instance, relationships might highlight the shared sentiment between different dialects, the influence of one dialect on another, or the presence of dialect-specific sentiment patterns.

By incorporating relationships within the sentiment ontology, sentiment analysis algorithms can leverage the structured knowledge to improve sentiment classification and extraction across multiple Arabic dialects. These relationships contribute to disambiguating sentiment, accommodating dialectal variations, and capturing the complexity and nuances of sentiment expression within the multi-dialects Arabic context.

In conclusion, constructing a multi-dialects Arabic sentiment ontology involves defin-

ing sentiment-related terms/concepts and establishing relationships to capture the variations and nuances of sentiment expression across different dialects. This structured approach enhances sentiment analysis accuracy and facilitates a deeper understanding of sentiment within the rich linguistic diversity of Arabic dialects.

In our work we kept the architecture of our ontology quite simple, we have splitted the main classes into two parts (MSA, Dialect), MSA class has subclasses which are (Negative, Positive), and Dialect class also has subclasses that are 25 cities that provided by the MADAR lexicon.

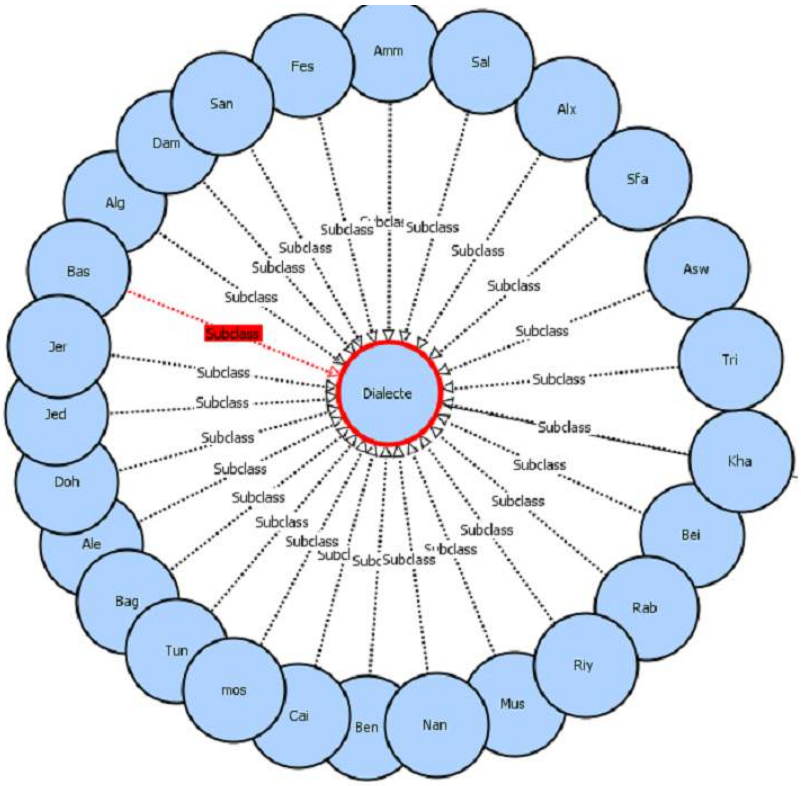


Figure 4.11: The Visual Notation for Dialect Class in Protege.

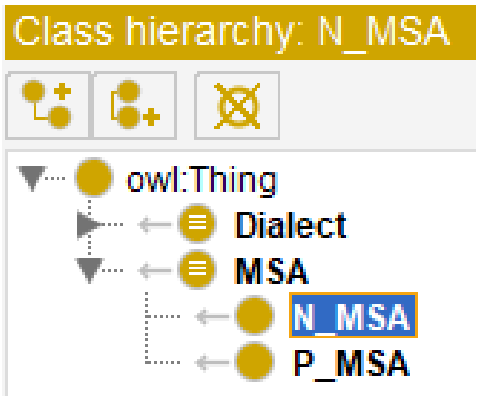


Figure 4.12: Hierarchy of the constructed ontology in the stem from using protege.

Within the MADAR lexicon, each dialect exhibits a fascinating phenomenon. Namely, there exists an MSA word that assumes distinct meanings across the various dialects. These MSA words carry polarities—negative, positive, or neutral. Remarkably, if an MSA word possesses a positive polarity, it subsequently imparts the same positive polarity to every connected word within the respective dialect.

By leveraging this simplified ontology structure, we can better understand the relationships between MSA words and their dialectical counterparts. This framework enables us to analyze sentiment polarities in the context of dialectal variations, shedding light on the intricate nuances of Arabic language usage.

MSA		Dialect			
Polarity	Word	Default	ISRI	Tashaphyne	City
1	قضّل	حب	حب	حب	SFA, ALG, TUN,
		خبر	خبر	خبر	ALG, TUN, SFA
		فضل	فضل	ضل	ALE, ALX, AMM, ASW, BAG, BAS, BEI, CAI, DAM, JED, JER, MOS, RIY, SAN, TRI, KHA, BEN, DOH, MUS, SAL, FES, RAB
		استخار	خار	استخار	MUS
		بدي	بدي	دي	ALE
		حبز	حبز	حبز	KHA
0	مريض	تعبان	تعب	عب	MUS, AMM, DOH, JED, JER, KHA, SAL
		عبان	عبن	عبي	KHA, FES, RAB
		علیل	علل	علیل	FES, SAN
		مالاد	الد	مالاد	ALG
		مریض	ریض	مریض	ALE, ALX, AMM, ASW, BAG, BAS, BEI, BEN, CAI, DAM, JED, JER, MOS, SAN, TRI, DOH, MUS, RIY, SAL, KHA, ALG, FES, RAB, SFA, TUN

Figure 4.13: Sample of information that can be extracted from the constructed ontology.

To accomplish this workflow, with the use of Protege software, we need to establish relationships between the concepts in our Ontology. Specifically, we will introduce a relationship called "synonym_of" that connects the domains (Dialect/ MSA) and ranges

of (Dialect/ MSA). Additionally, we will assign the characteristics of "Symmetric" and "Reflexive" to this relationship.

The "synonym_of" relationship enables us to denote that certain concepts in the Dialect and MSA domains share similar meanings or can be used interchangeably. For example, if two words from different dialects or between MSA and a dialect have equivalent or closely related meanings, we can establish a synonym relationship between them.

By assigning the characteristics of "Symmetric" and "Reflexive" to the "synonym_of" relationship, we ensure that the relationship is bidirectional and self-reflective. This means that if Word A is a synonym of Word B, then Word B is also a synonym of Word A, and each word is considered a synonym of itself.

This approach allows us to capture the complex and dynamic nature of language and enables us to navigate the ontology to identify synonymous concepts across dialects and MSA. It forms the foundation for analyzing sentiment and understanding the shared meanings between different linguistic variations.

By incorporating these relationships and characteristics into our ontology, we can unlock valuable insights and enhance our understanding of the relationships and semantic connections within the Arabic language landscape.

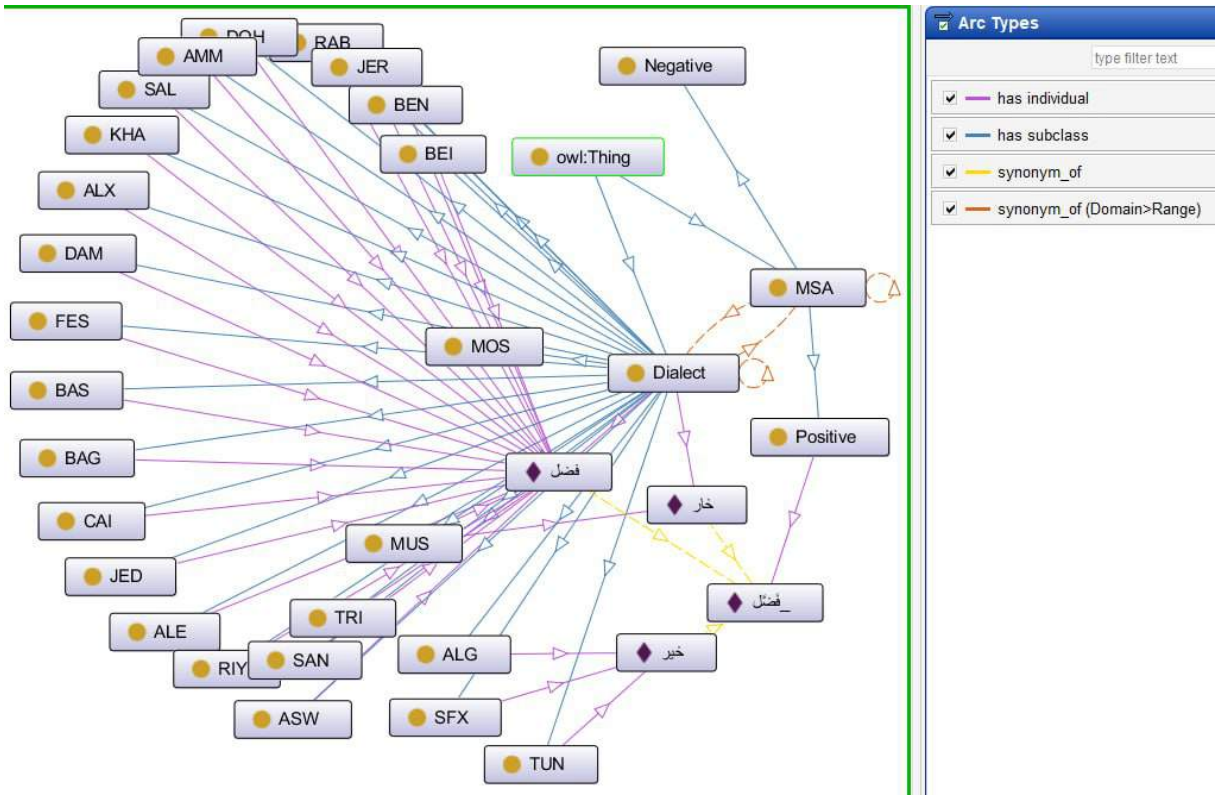


Figure 4.14: An example of the MSA word "**فضل**" and its Arabic multi-dialects synonyms using ISRI stemmer.

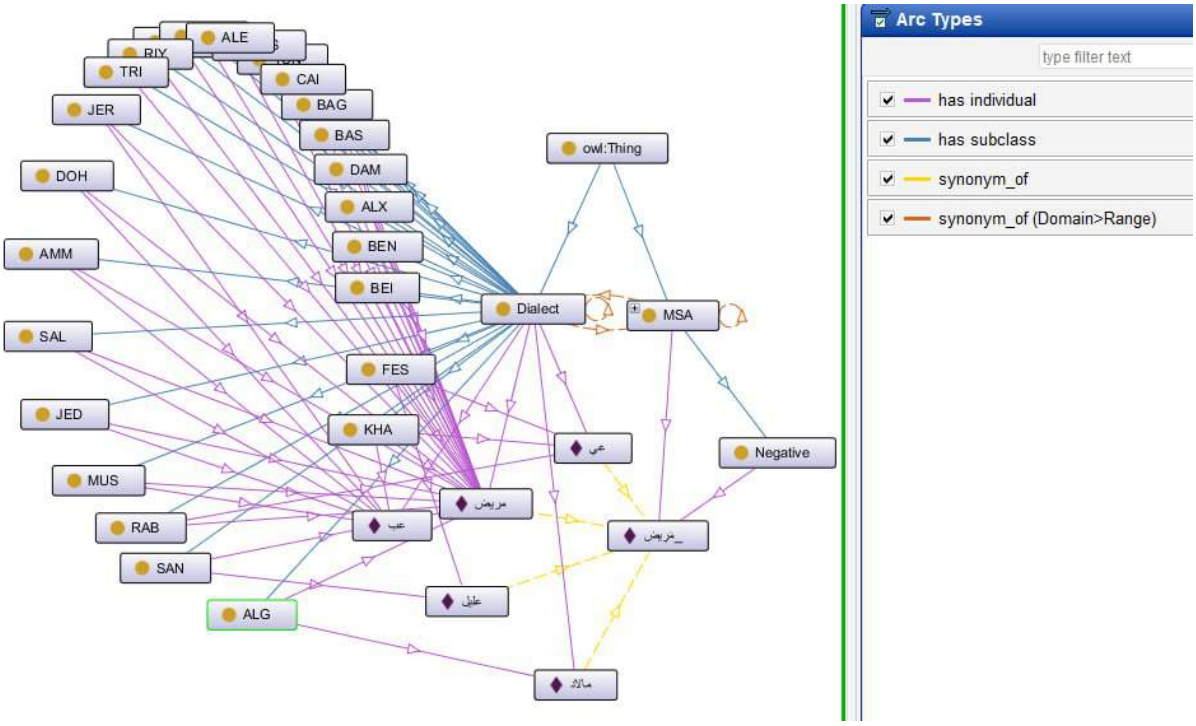


Figure 4.15: An example of the MSA word "**مريض**" and its Arabic multi-dialects synonyms using Tashaphyne stemmer.

Note: this ontology filled with two different data terms (Tashaphyne, Isri).

4.3 Sentiment analysis

Basically it's the process of combining the ontology that we are created and the pre-processed data of **HARD Hotel Review**, then try to classify each review if it's positive, negative or neutral, the classifying process based on:

- Taking each word from HARD Hotel Review.
- Using the owlready2 library in Python, we search for the word within our ontology.

```
from owlready2 import *
# Load the ontology
onto1 = get_ontology("onto.owl").load()

def Onto_search(ontology, instence):
    try:
        existed_instence =  ontology.search(iri =str("*"+ instence)) # Searching for the giving word
        if(str(existed_instence) != 'None'): # if does exist
           return [{"inctence":existed_instence,"synonym_of":existed_instence[0].synonym_of,
                    "dialect_synonyms": ontology.search( synonym_of = existed_instence[0].synonym_of[0]),
                    "polarity":existed_instence[0].synonym_of[0].is_a}]
        else: # if doesn't exist
            return[] #
    except:          # the return empty list refers to the given word polarity is count as neutral
        return[]     #
```

Figure 4.16: Python search method for a word in the ontology.

- Once we find the word in the ontology, we extract its associated polarity.
- If a word cannot be found in the ontology, assigning a default polarity value of two to indicate unknown or undefined sentiment.

```
test = Onto_search(onto1, "جميل")
test
✓ 0.0s
[{'inctence': [onto.جميل],
  'synonym_of': [onto._جَمِيل, onto._جَيِّد, onto._رائع],
  'dialect_synonyms': [onto.جميل, onto.حليوه, onto.وسيم],
  'polarity': [onto.Positive]}]
```

Figure 4.17: Extracting the polarity of MSA word "**جميل**" from the ontology.

- Compare the counts of positive, negative polarities within the review.
- Determine the polarity category with the highest count. If there is a tie, choose Two as the polarity.

```
tokens = Hard_Review_Tokens_list # One Review
polarity_list = []
for token in tokens:
    try:
        msa_word = Onto_search(Ontology,token)
        token_polarity = msa_word["polarity"]
        polarity_list.append(token_polarity)
    except:
        polarity_list.append(2) # if the token was not found in Ontology
                                # token will classify as neutral (2)
if polarity_list.count(0) > polarity_list.count(1):
    print("polarity of this Review is :", 0)
elif polarity_list.count(0) == polarity_list.count(1):
    print("polarity of this Review is :", 2)
else:
    print("polarity of this Review is :", 1)
```

Figure 4.18: Polarity calculation using scores.

- Assigning the most replicated polarity as the overall sentiment of the review.

4.4 Polarity Calculation

كويس	العمل	وطاقم	سريعه	خدمة	كثير	منيح	الفندق
positive	neutral	neutral	positive	neutral	neutral	positive	neutral

Polarity Calculation:

- Positive: 3 (كويس, سريعه, منيح)
- Neutral: 5 (العمل, وطاقم, خدمة, كثير, الفندق)
- Negative: 0

Positive > Negative

Polarity = Positive

Figure 4.19: Example about calculating the polarity of the HARD review.

Since the count of positive words is greater than the count of negative words, we can conclude that the overall polarity of the review is positive.

Stemming is a common technique used in natural language processing to reduce words to their base form. However, when applied to Arabic dialects, it faces challenges. The existing stem tools used in this project, designed for Modern Standard Arabic (MSA), can lead to inaccuracies and misinterpretations when applied to dialectal words.

The stemming-based sentiment analysis system in this project acknowledges the limitations of the stem tools for handling Arabic dialects. Dialectal words processed using these tools may produce stem outputs that deviate from the intended meaning or fail to capture the correct semantics. To address these challenges, a second model is introduced in the next section. This model focuses on extracting similarity between words and the vocabulary in our ontology. By leveraging machine learning techniques, it aims to provide a more accurate and context-aware sentiment analysis process.

The second model takes into account the specific lexical nuances and semantic variations in Arabic dialects. By considering the contextual meaning of words and their connections to the ontology, it aims to improve sentiment analysis outcomes.

By introducing this enhanced model, the goal is to enhance the accuracy and effectiveness of sentiment analysis for Arabic dialects, overcoming the limitations of traditional stemming approaches.

5 Machine learning-based sentiment analysis system

Sentiment analysis has become increasingly important in understanding customer sentiments and improving services in various industries. However, sentiment analysis in Arabic dialect text presents unique challenges due to the regional variations and nuances present in different dialects. This project aims to address these challenges by developing a sentiment analysis system specifically for Arabic dialect hotel reviews. By leveraging the power of ontology-based representation and deep learning techniques, we can accurately analyze the sentiment of hotel reviews and gain valuable insights into customer opinions. In Figure 4.20 below our proposed model.

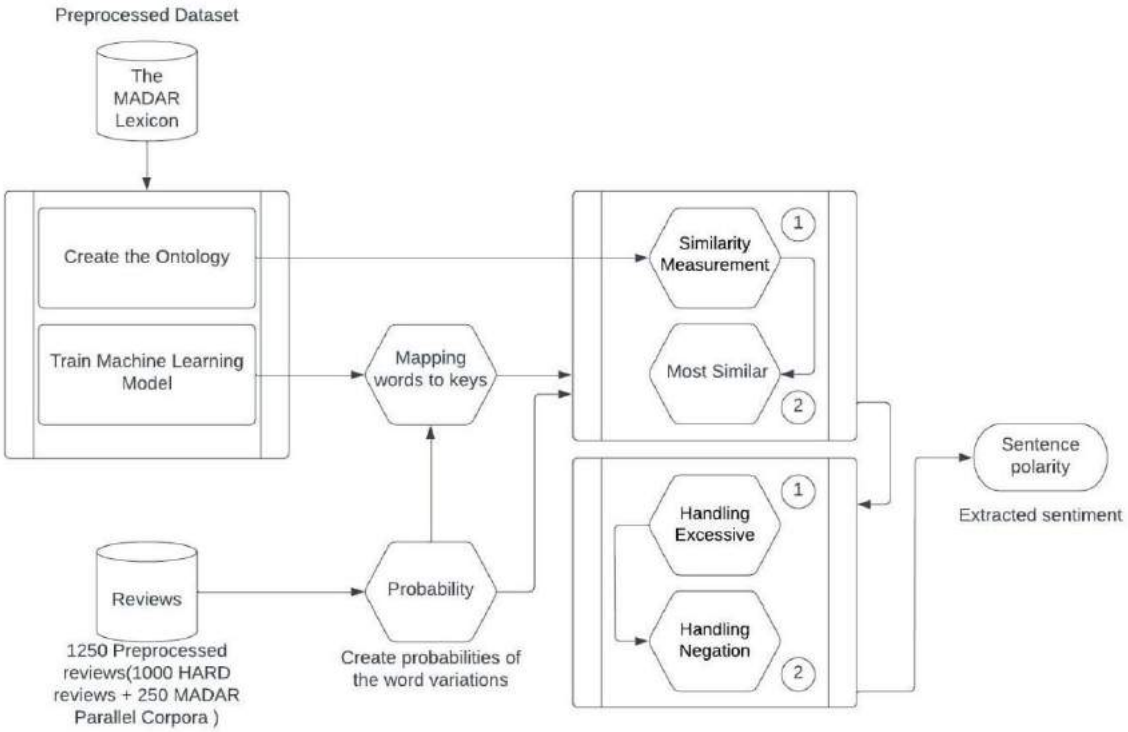


Figure 4.20: Machine learning-based sentiment analysis system Architecture.

5.1 Converting dialect words to numeric representation and key encoding

5.1.1 Converting dialect words to numeric representation

To facilitate the training of the RNN+LSTM model, dialect words are transformed into a numeric representation. This process involves assigning a unique numerical value to each character or symbol present in the dialect words. This step is essential for training the model and establishing the correspondence between dialect words and their corresponding MSA words. The model can effectively process the input data and learn patterns and relationships between the numerical representations of dialect words.

The function (convert_dialect_to_numeric) in following Figure will explain how this process does work:

```
+ Code  + Markdown

def convert_dialect_to_numeric(dialect_word, arabic_letters ):# arabic_letters is CSV file contains
    numeric_list = []                                         # the Arabic letters
    for char  in dialect_word :
        numeric_value  = 0
        for i in range(0,len(arabic_letters)-1):              # The letters will coded based on
            if (arabic_letters.iloc[i]['chars'] == char ) :   # theirs index ordering in the CSV
                numeric_value = i+1
        numeric_list.append(numeric_value)
    numeric_list
    filtered_list  =  list(filter(lambda num: num != 0, numeric_list))
    b = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    numeric_list.extend(b[len(filtered_list):])
    numeric_array  = np.array([numeric_list])
    return numeric_array
✓ 0.0s

convert_dialect_to_numeric("جميلين", arabicLatter)
✓ 0.0s
array([[ 5, 24, 28, 23, 28, 25,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0,  0,  0,  0]])
```

Figure 4.21: Code python of Converting Dialect Words to Numeric and test example.

5.1.2 KEY Encoding

In addition to converting the dialect words, each word is tagged with a unique key. This key serves as an identifier that corresponds to the MSA word in the ontology that aligns with the given dialect word. By tagging the words with keys, we establish a mapping between the dialect words and their corresponding MSA words, enabling accurate sentiment analysis based on the sentiment expressed in the MSA word.

By performing these data processing steps, we ensure that the dataset is appropriately prepared for training the RNN+LSTM model and establishing the dialect-to-MSA mapping.

5.2 Multi-dialect Arabic ontology construction

The model begins with the construction of an ontology, which serves as a structured representation of the Arabic language. The ontology architecture includes classes such as MSA, Dialect, Positive, Negative, Excessive, and Negation. The MSA class acts as the superclass for subclasses like Positive, Negative, Excessive, and Negation, enabling a hierarchical representation of sentiment-related words.

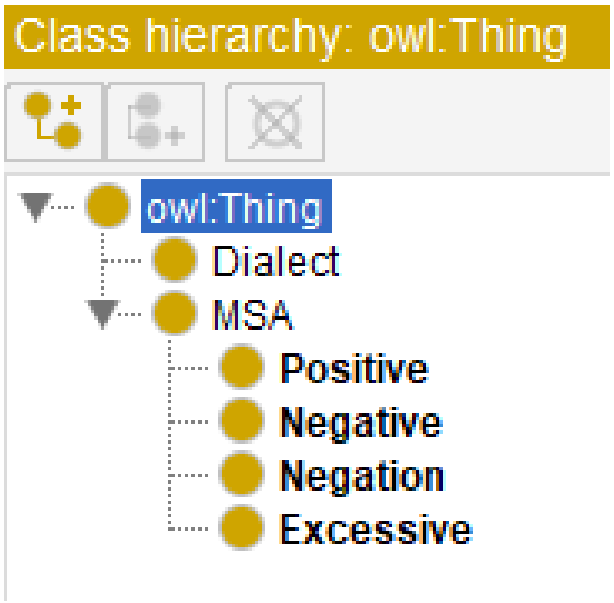


Figure 4.22: hierarchy of the constructed ontology using dialectal words.

The Dialect class serves as a superclass for multiple subclasses, representing specific Arabic dialects spoken in various regions or cities. Each dialect subclass captures the vocabulary and linguistic characteristics unique to that dialect, allowing for accurate sentiment analysis based on regional variations.

The ontology incorporates the "synonym_of" relationship, which allows the mapping of dialect words to their corresponding MSA words. Additionally, the MSA class contains a data property named "key" that uniquely identifies each MSA word, aiding in efficient retrieval and analysis.

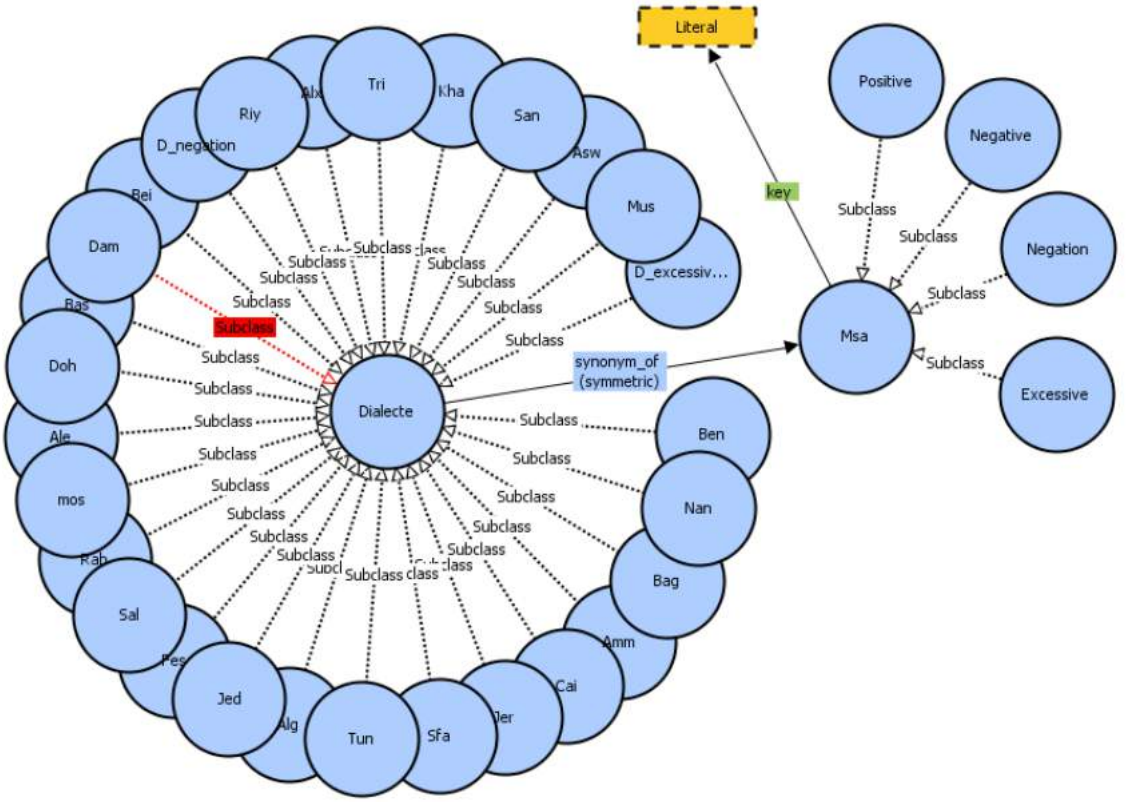


Figure 4.23: The Visual Notation for Ontology in Protege.

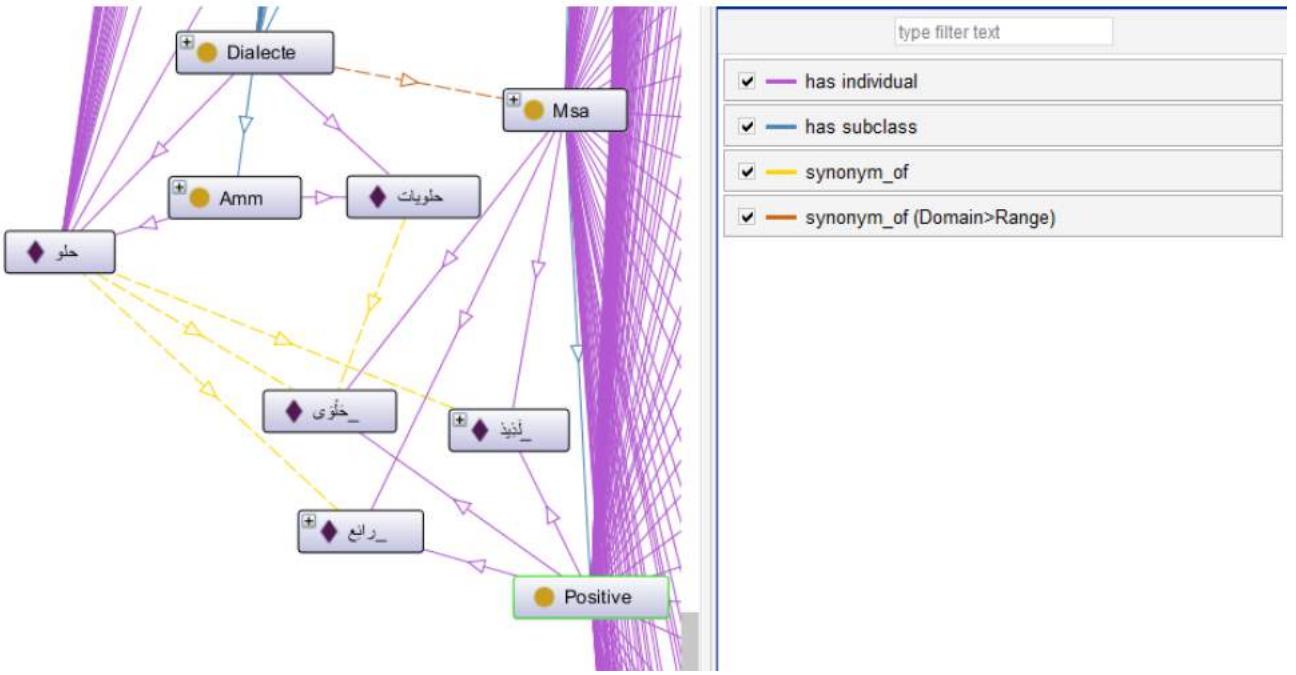


Figure 4.24: different synonyms of the MSA word"**لذيذ**".

5.3 Enhanced Sentiment Analysis with RNN+LSTM Model

Dialect-mainstream language mapping is a challenging task that requires an effective model to measure the similarity between dialect words and mainstream language (MSA) words. The RNN+LSTM model is a powerful solution that addresses this problem by capturing patterns and relationships between the two languages. It has proven to be highly effective in various natural language processing tasks. Let's explore the key aspects and benefits of using the RNN+LSTM model in language-related applications.

5.3.1 Benefits and Applications

- **Capturing Dialect-Mainstream Language Mapping:** The RNN+LSTM model excels at capturing the mapping between dialect words and their corresponding MSA words. During the training process, it learns the underlying patterns and relationships between the two languages. Then, during inference, given a dialect word, the model predicts the most similar MSA word based on the learned representations. This capability makes it a valuable tool for dialect-mainstream language translation, sentiment analysis, and other language-related tasks.

```
Input Dialect Word: وحلو
The input sequence : [[27  6 23 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]]
Predicting...
1/1 [==============================] - 0s 23ms/step
Predicted Key: 340
```

	MSA	Dialect	CODA	Key
15405	حَلْوَى	ALE	تحلايه	340
15406	حَلْوَى	DAM	تحلايه	340
15407	حَلْوَى	BEI	تحلايه	340
15408	حَلْوَى	SAN	حالي	340
15409	حَلْوَى	MUS	حرور	340
15410	حَلْوَى	BAG	حلاوه	340
15411	حَلْوَى	BAS	حلاوه	340
15412	حَلْوَى	MOS	حلاوه	340
15413	حَلْوَى	MUS	حلاوه	340
15414	حَلْوَى	DOH	حلو	340
15415	حَلْوَى	KHA	حلو	340
15416	حَلْوَى	AMM	حلو	340
15417	حَلْوَى	BEN	حلو	340

Figure 4.25: Mapping example between the dialect word and its corresponding.

- **Efficiency and Scalability:** One of the notable advantages of the RNN+LSTM

model is its efficiency and scalability. Unlike traditional methods that require iterating through the entire ontology to measure similarity, the model encodes the input dialect word using its trained embedding layer. It then generates predictions directly, saving computational time and resources. This efficiency allows for real-time or batch processing of large volumes of dialect word reviews, making it suitable for applications with high demands.

- **Generalization and Adaptability:** The RNN+LSTM model demonstrates good generalization capabilities, enabling it to predict the most similar MSA word even for dialect words that were not present in the training data. This ability is attributed to its capacity to capture semantic information and patterns from the training dataset. Moreover, the model can be fine-tuned or expanded with additional data or features, allowing it to adapt to specific requirements or changes in the language domain. This flexibility makes it a robust and adaptable solution for various language-related tasks.
- **Model Architecture:** Here is the visual representation of the RNN+LSTM model architecture:

```
# Define the maximum sequence length and vocabulary size
max_sequence_length = 20    # Maximum length of dialect words
vocabulary_size = 972       # Number of unique MSA words

# Define the model architecture
model = Sequential()
model.add(Embedding(vocabulary_size, 100, input_length=max_sequence_length))

model.add(Masking(mask_value=0.0))  # Masking layer to handle variable-length input

model.add(Bidirectional(LSTM(150, return_sequences=True)))
model.add(Dropout(0.2))

model.add(Bidirectional(LSTM(100, return_sequences=False)))
model.add(Dropout(0.2))

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(vocabulary_size, activation='softmax'))

# Compile the model
adam = Adam(learning_rate=0.001)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
```

Figure 4.26: The visual representation of the RNN+LSTM model architecture.

The model consists of the following layers:

- **Embedding layer:** Converts the input dialect word into a dense vector representation of size 100.

- **Masking layer:** Handles variable-length input by masking padded values (0.0).

- **Bidirectional LSTM layers:** Captures the sequential information in both forward and backward directions. The first LSTM layer has 150 units and returns sequences, while the second LSTM layer has 100 units and returns only the final output.

- **Dropout layers:** Regularize the model by randomly dropping out a fraction of inputs during training to prevent over fitting. Dropout rate is set to 0.2.

– **Dense layers:** Fully connected layers with relu activation function. The first dense layer has 512 units, the second dense layer has 256 units, and the final dense layer has vocabulary_size units with softmax activation, representing the probability distribution over the vocabulary.

- **Training Process:**

The model was trained on the entire dataset, which consisted of 42,196 rows. This approach was taken to ensure that the model learns from a comprehensive representation of the dialect words and their corresponding MSA word representations. By using the entire dataset, the model is exposed to a wide range of variations and mappings, enhancing its ability to capture patterns and relationships. The objective of this training was to enable the model to learn patterns and representations that would facilitate measuring similarity between dialect words and their corresponding MSA word representations.

The training process involved iterating over the dataset for 200 epochs, optimizing the model's parameters using the Adam optimizer with a learning rate of 0.001. The model was trained using the categorical cross-entropy loss function to encourage the learning of meaningful representations for similarity measurement.

During training, the model was exposed to various dialect words and their MSA word representations. By learning from the entire dataset, the model was able to capture the diverse patterns and relationships present in the data, enhancing its ability to measure similarity effectively.

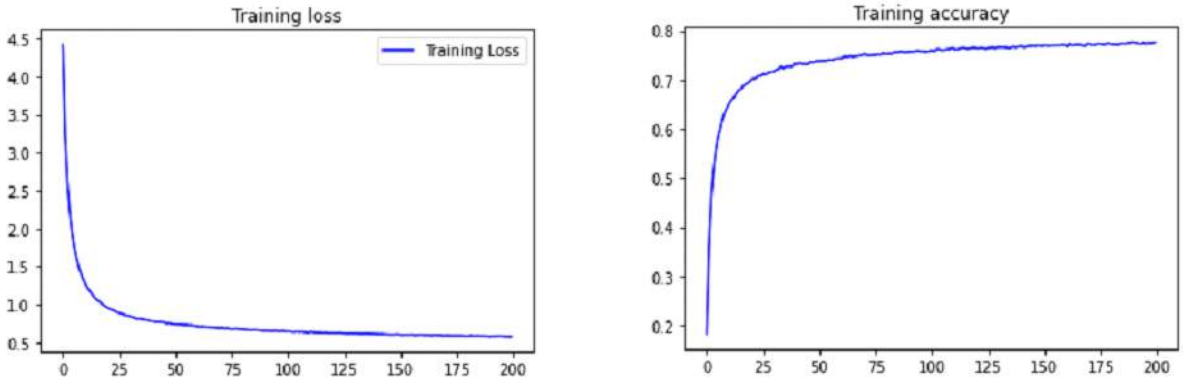


Figure 4.27: The final training statistics after the 200th epoch.

These values (Loss=0.55%,Accuracy=0.77%) indicate the model's performance on the training dataset after the completion of the training process.

The performance evaluation in this case can be based on the model's ability to accurately measure similarity between dialect words and MSA word representations within the training process itself. However, it is important to note that the model's performance on unseen data or in real-world scenarios may still vary, and it is recommended to validate its effectiveness on new and independent datasets, if available.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 20, 100)           97200
_________________________________________________________________
masking (Masking)            (None, 20, 100)           0
_________________________________________________________________
bidirectional (Bidirectional (None, 20, 300)           301200
_________________________________________________________________
dropout (Dropout)            (None, 20, 300)           0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 200)               320800
_________________________________________________________________
dropout_1 (Dropout)          (None, 200)               0
_________________________________________________________________
dense (Dense)                (None, 512)               102912
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 972)               249804
=================================================================
Total params: 1,203,244
Trainable params: 1,203,244
Non-trainable params: 0
```

Figure 4.28: the summary of the model architecture.

5.4 Handling Excessive Words

In sentiment analysis, the accurate interpretation of text is crucial to capture the intended sentiment expressed by the author. However, certain words can have a unique impact on sentiment analysis and require special attention. These words, referred to as

"excessive words" possess the potential to influence the sentiment polarity of the surrounding words in a review. To address this, a specific function has been implemented to handle excessive words within our sentiment analysis system. This section explores the approach used to handle excessive words and its benefits in improving sentiment analysis results.

5.4.1 Understanding Excessive Words

Excessive words, in the context of our sentiment analysis system, are words that exert a significant influence on the sentiment polarity of adjacent words. These words have been identified and classified within our ontology due to their distinct impact on sentiment analysis.

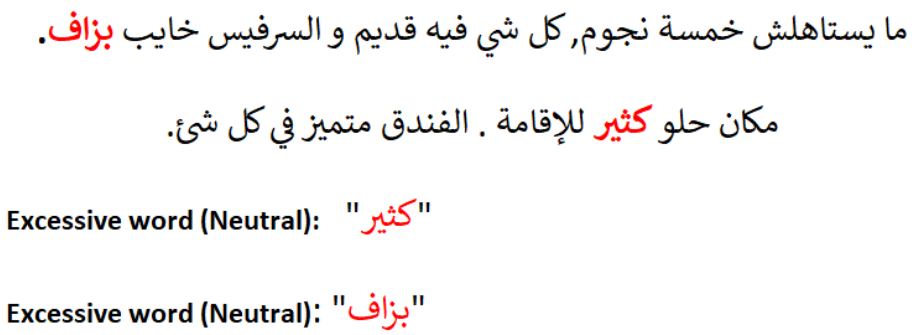


Figure 4.29: example of excessive words in dialectal form.

5.4.2 The Contextual Approach

To ensure accurate sentiment analysis, our system adopts a contextual approach when encountering excessive words. When an excessive word is identified within a review, its polarity is determined based on the polarity of the previous word. By considering the sentiment polarity of the preceding word, we aim to assign the same polarity to the excessive word, capturing the intended sentiment more accurately.

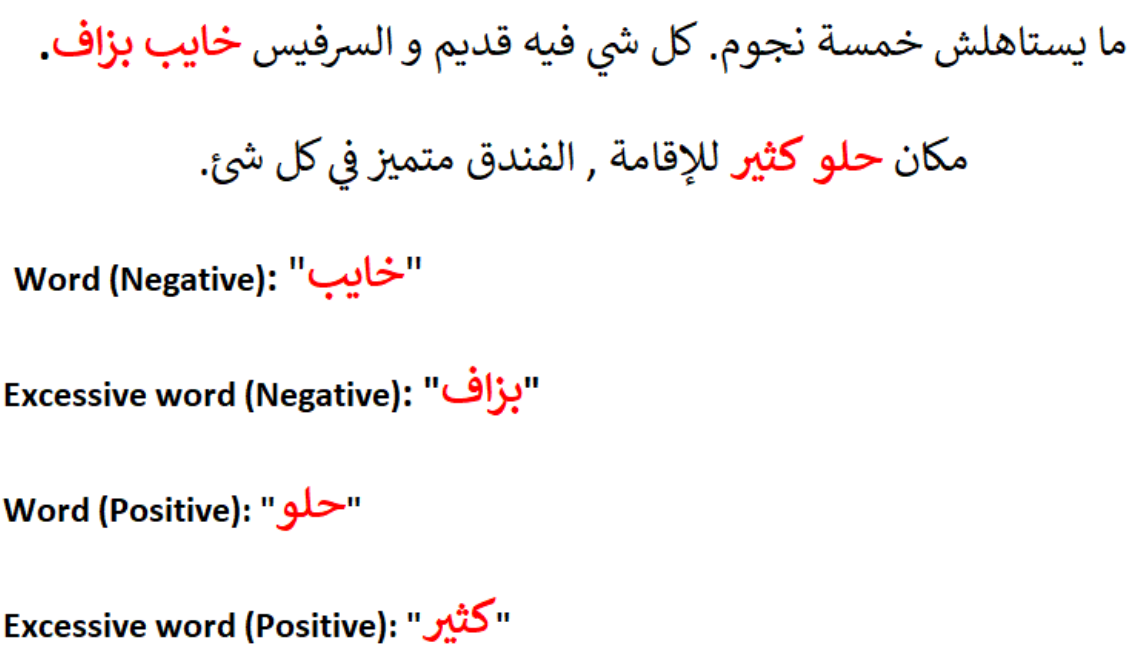


Figure 4.30: Assigning different polarities to excessive words.

5.4.3 Algorithm for Handling Excessive Words

To implement this contextual approach, we have developed a specific algorithm. The algorithm follows these steps:

- Input: Tokenized review (list of words).
- Output: Modified review with updated polarities for excessive words.
- Initialize an empty list to store the modified review.
- Initialize a variable to store the polarity of the previous word (initialized as neutral).
- Iterate through each word in the tokenized review:
  - Check if the word belongs to the "Excessive" class in the ontology.
    - If the word is identified as an excessive word:
      Assign the polarity of the previous word to the excessive word.
  - Append the word to the modified review list.
  - Return the modified review with updated polarities for excessive words.

5.4.4 Enhancing Sentiment Analysis Results

By incorporating the handling of excessive words, our sentiment analysis system improves the accuracy and contextual understanding of sentiment in Arabic dialect reviews. This approach ensures that excessive words inherit the sentiment polarity of the preceding word, capturing the intended sentiment more effectively.

5.4.5 Customization and Adaptation

It is important to note that the identification and handling of excessive words are based on our specific ontology and the requirements of our sentiment analysis system. The algorithm can be customized and adapted to accommodate the unique linguistic characteristics and nuances of different languages or dialects.

5.5 Handling Negation Words

Negation words play a crucial role in language, as they have the power to reverse the sentiment or meaning expressed in a sentence. When performing sentiment analysis or any form of text analysis, it is essential to consider the influence of negation words on the overall interpretation of the text. This section explores the significance of handling negation words in sentiment analysis and discusses an approach to effectively incorporate them into the analysis process.

5.5.1 Understanding Negation Words

Negation words are a specific class of words that negate or deny the truth, existence, or occurrence of something in a sentence. They indicate a negative sentiment or reverse the polarity of the expressed sentiment. Example of negation words Figure 1.4. These words have a profound impact on the sentiment conveyed by a sentence.

5.5.2 The Impact of Negation on Sentiment Analysis

Negation words can significantly alter the sentiment or polarity of a sentence. Consider the following example:

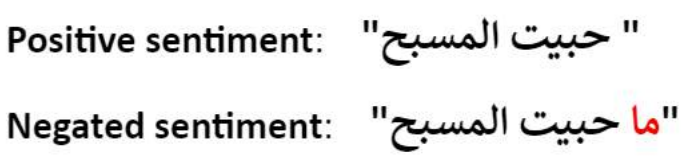


Figure 4.31: The effect of negation on changing the polarity.

In the negated sentiment example, the inclusion of the negation word, completely reverses the positive sentiment expressed in the original statement. Negation words have the potential to change the overall sentiment from positive to negative or vice versa.

5.5.3 The Need to Handle Negation Words

To ensure accurate sentiment analysis, it is crucial to handle negation words appropriately. Ignoring or overlooking negation words can lead to inaccurate sentiment assessments

and misinterpretations of the text. By incorporating negation word handling into sentiment analysis algorithms, we can obtain a more context-aware understanding of sentiment in text.

5.5.4 Algorithm for Handling Negation Words

To handle negation words effectively, the following algorithm can be employed within sentiment analysis systems:

- Input: Tokenized review (list of words).
- Output: Modified review with updated polarities for negation words.
- Initialize an empty list to store the modified review.
- Initialize a variable to store the polarity of the previous word (initialized as neutral).
- Iterate through each word in the tokenized review:
  - Check if the previous word belongs to the quot;Negationquot; class in the ontology.
  - If the previous word is identified as a negation word:
    - If the current word has a positive polarity, update its polarity to negative.
    - If the current word has a negative polarity, update its polarity to positive.
  - Append the word to the modified review list.
- Return the modified review with updated polarities for negation words.

By applying this algorithm, the sentiment analysis system can accurately account for the effect of negation on the sentiment polarity of words within a text.

5.5.5 Benefits of Handling Negation Words

Handling negation words in sentiment analysis brings several benefits:

- **Improved accuracy:** By considering the impact of negation, sentiment analysis algorithms can provide more accurate assessments of sentiment in text.
- **Context-aware analysis:** Incorporating negation handling allows for a contextually sensitive interpretation of sentiment, capturing the intended meaning behind negated statements.
- **Enhanced understanding:** By accounting for negation, sentiment analysis systems can better comprehend complex opinions and sentiments expressed in text.

5.6 The Probability-Driven Similarity Measurement Approach

In the field of sentiment analysis, accurately capturing the sentiment expressed in user-generated content is crucial for understanding public opinion and making informed decisions. However, sentiment analysis for Arabic dialect reviews poses unique challenges due to the informal nature of the language and variations in dialects. To address these challenges, we propose the Probability-Driven Similarity Measurement Approach, a novel technique that combines probabilistic methods with similarity measurement to enhance the accuracy of sentiment analysis in Arabic dialect reviews. This section delves into the details of this approach, including its algorithmic implementation, and its application in sentiment analysis.

5.6.1 Probability-Driven Variation Generation

The first step in the Probability-Driven Similarity Measurement Approach is to generate variations of the dialect word. This process aims to account for potential misspellings or dialectal variations that exist in the reviews. By systematically removing letters from the original word, we create a range of possibilities for each word.

```
for counter in range(9):
    current_variation = ""
    if (counter > 5) and (counter < 9):
        if (counter - 2 > len(dialect_word)):
            break
        else:
            if (len(dialect_word[1:-counter + 5]) >= 2):
                current_variation = dialect_word[1:-counter + 5]
            else:
                break
    elif (counter > 1) and (counter <= 5):
        if (counter > len(dialect_word) - 1):
            break
        else:
            current_variation = dialect_word[:-counter + 1]
    elif counter == 1:
        if len(dialect_word) <= 2:
            break
        else:
            current_variation = dialect_word[1:]
    else:
        if counter == 0:
            current_variation = dialect_word
        else:
            break
```

Figure 4.32: Algorithm of Probability-Driven Variation Generation.

5.6.2 Explanation

This algorithm generates variations of the dialect word by systematically removing letters from the original word. It iterates through a range of values from 0 to 8, and based on the value of the counter, it determines which letters to remove from the dialect word to create the variation. The generated variations are stored in the current_variation variable for further processing in the sentiment analysis pipeline.

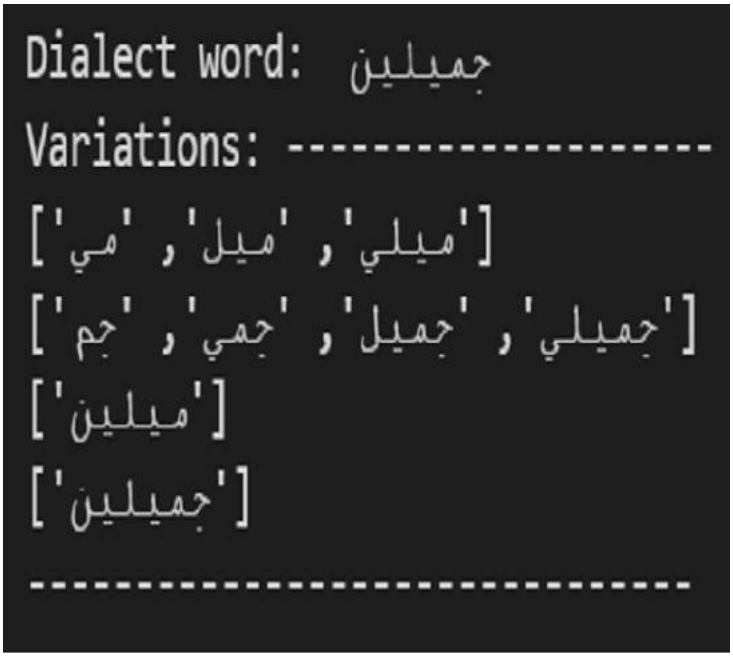


Figure 4.33: The resulting variations of the dialectal word "**جميلين**" after applying the probability.

5.6.3 Mapping to MSA Words

We have used the constructed ontology of MSA/dialect words mentioned in Section (5.2) that associates dialect words with their corresponding MSA counterparts. Although the generated variations may not typically exist in our ontology, they are generated specifically for the purpose of similarity measurement. Each variation is matched with its corresponding MSA word from the ontology using a machine learning model (RNN+LSTM model) mentioned in Section (6.3). This model enables us to predict the most suitable MSA word for each generated variation, allowing us to establish accurate connections between the dialect word variations and their standardized equivalents in our ontology.

Corresponding (MSA) to: حلوين

	MSA	Dialect	CODA	Key
12489	مَبْنَى	ALG	باتيمو	769
12493	مَبْنَى	SAN	بنا	769
12494	مَبْنَى	ASW	بنايه	769
12513	مَبْنَى	MUS	بنيان	769
12514	مَبْنَى	SFA	بنيه	769
12516	مَبْنَى	FES	موبل	769
12518	مَبْنَى	ALE	عماره	769
12537	مَبْنَى	ALX	مبني	769

Figure 4.34: Mapping the dialectal word to its MSA word "**حلوين**" using the ontology.

Corresponding (MSA) to: لوين

	MSA	Dialect	CODA	Key
30279	إسْهال	ALE	اسهال	127
30302	إسْهال	ALG	دياري	127
30303	إسْهال	ALX	ليونه	127

Figure 4.35: Mapping the dialectal word to its MSA word "**لوين**" using the ontology.

Corresponding (MSA) to: حلوي

	MSA	Dialect	CODA	Key
15405	حَلْوَى	ALE	تحلايه	340
15408	حَلْوَى	SAN	حالي	340
15409	حَلْوَى	MUS	حرور	340
15410	حَلْوَى	BAG	حلاوه	340

Figure 4.36: Mapping the dialectal word to its MSA word "**حلوي**" using the ontology.

Corresponding (MSA) to: حلو

	MSA	Dialect	CODA	Key
20678	جَمِيل	DOH	جميل	302
20696	جَمِيل	SAN	حالي	302
20697	جَمِيل	DOH	حلو	302
20716	جَمِيل	JED	حليوه	302

Figure 4.37: Mapping the dialectal word to its MSA word "**حلو**" using the ontology.

Corresponding (MSA) to: حل

	MSA	Dialect	CODA	Key
30441	إِلْغاء	AMM	ازاله	136
30444	إِلْغاء	KHA	الغا	136
30445	إِلْغاء	ALE	الغاء	136
30469	إِلْغاء	ALG	انولاسيون	136

Figure 4.38: Mapping the dialectal word to its MSA word "**حل**" using the ontology.

5.6.4 Similarity Measurement

Once the dialect word variations are mapped to MSA words, we employ similarity measurement techniques to quantify the similarity between each variation and the known dialect words associated with the MSA word. Techniques such as word embedding or similarity measures like Word2Vec can be used to calculate the similarity scores.

Corresponding (MSA) to: حلـويـن

	MSA	Dialect	CODA	Key	similarity
12489	مَبْنَى	ALG	باتيمو	769	0.027057
12493	مَبْنَى	SAN	بنا	769	0.092917
12494	مَبْنَى	ASW	بنايه	769	0.016135
12513	مَبْنَى	MUS	بنيان	769	-0.059876
12514	مَبْنَى	SFA	بنيه	769	-0.027750
12516	مَبْنَى	FES	موبل	769	-0.111671
12518	مَبْنَى	ALE	عماره	769	-0.052347
12537	مَبْنَى	ALX	مبني	769	-0.010839

Figure 4.39: The highest similarity score between the dialectal word "**حلوين**" (variation) and the proposed corrections.

Corresponding (MSA) to: لـويـن

	MSA	Dialect	CODA	Key	similarity
30279	إسْهال	ALE	اسهال	127	-0.111671
30302	إسْهال	ALG	دياري	127	-0.052347
30303	إسْهال	ALX	ليونه	127	-0.010839

Figure 4.40: The highest similarity score between the dialectal word "**لوين**" (variation) and the proposed corrections.

Corresponding (MSA) to: حلوي

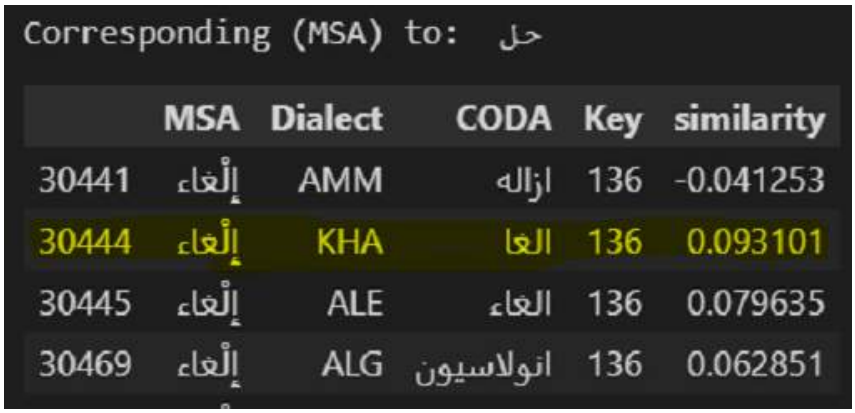
	MSA	Dialect	CODA	Key	similarity
15405	حَلْوَى	ALE	تحلايه	340	0.216171
15408	حَلْوَى	SAN	حالي	340	0.027057
15409	حَلْوَى	MUS	حرور	340	0.092917
15410	حَلْوَى	BAG	حلاوه	340	0.016135
15414	حَلْوَى	DOH	حلو	340	-0.059876

Figure 4.41: The highest similarity score between the dialectal word "**حلوي**" (variation) and the proposed corrections.

Corresponding (MSA) to: حلو

	MSA	Dialect	CODA	Key	similarity
20678	جَمِيل	DOH	جميل	302	0.054334
20696	جَمِيل	SAN	حالي	302	-0.037720
20697	جَمِيل	DOH	حلو	302	1.000000
20716	جَمِيل	JED	حليوه	302	-0.041253

Figure 4.42: The highest similarity score between the dialectal word"**حلو**" (variation) and the proposed corrections.



Corresponding (MSA) to: حل

	MSA	Dialect	CODA	Key	similarity
30441	إِلْغاء	AMM	ازاله	136	-0.041253
30444	إِلْغاء	KHA	الغا	136	0.093101
30445	إِلْغاء	ALE	الغاء	136	0.079635
30469	إِلْغاء	ALG	انولاسيون	136	0.062851

Figure 4.43: The highest similarity score between the dialectal word "**حل**" (variation) and the proposed corrections.

5.6.5 Selection of Best Variation

Based on the similarity scores obtained, we select the variation that exhibits the highest

similarity to the known dialect words. This selected variation represents the most likely intended dialect word, considering potential misspellings or variations. From the previous figures that we saw before, the version of Figure 4.42 it has the maximum similarity among the entire variations of the given word, Therefore it will be the best variation to select for the word Figure (4.39).

5.6.6 Polarity Assignment

With the selected variation determined, we retrieve the polarity of the corresponding MSA word from the ontology. The polarity of the selected variation is assigned based on the polarity associated with the MSA word. This step enables us to determine the sentiment of the dialect word, considering its variations and similarities to the known dialect words.

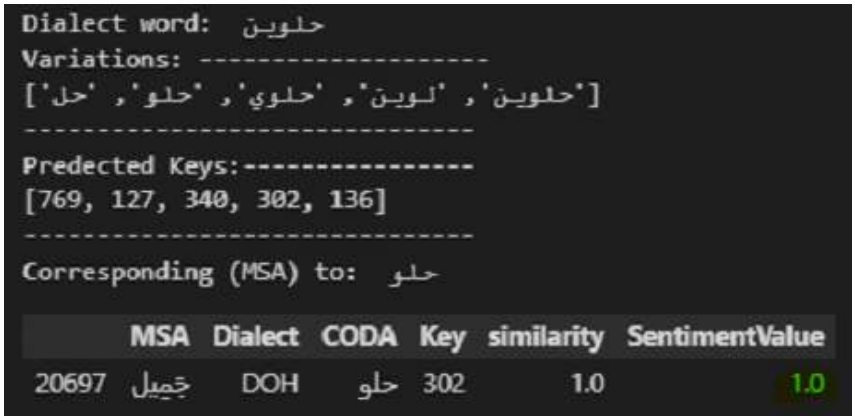


Figure 4.44: The polarity of the dialectal word "**حلوين**" after performing the probability-driven similarity measurement.

The Probability-Driven Similarity Measurement Approach is a powerful technique that enhances the accuracy of sentiment analysis in Arabic dialect reviews. By combining probabilistic methods with similarity measurement, we effectively handle variations and nuances in dialect words, leading to more accurate sentiment classification results.

5.7 Sentiment Analysis

Basically it's the process of combining the Ontology that we are created and the pre-processed data of (1000 HARD Hotel Arabic Reviews Dataset + 250 The MADAR Arabic Dialect Corpus Dataset), then try to classify each review if it's positive, negative or neutral, the classifying process based on:

- Taking each word from (1000HARD Hotel Review dataset + 250 the MADAR Arabic Dialect corpus Dataset).

- Tokenize the sentence.
- Perform probability-driven variation generation.
- Convert each word variation to a 20-dimensional vector.
- Feed the vectors into an RNN+LSTM model and predict the key number.
- Find the key in the ontology and extract the corresponding MSA word.
- Find the most similar variation.
- Handle negation and excessive words.
- Perform polarity assignment.
- Calculate the polarity of the sentence based on word polarities.

```
sentence = "هذا مثال"

tokens = tokenize_sentence(sentence)

word_variations = [generate_word_variations(token) for token in tokens]

vectors = [convert_word_to_vector(variation) for variation in word_variations]

key_number = predict_key_number(vectors)

msa_word = find_msa_word(key_number)

most_similar_variation = find_most_similar_variation(word_variations)

handled_sentence = handle_negation_excessive_words(sentence)

word_polarities = perform_polarity_assignment(handled_sentence)

sentence_polarity = calculate_sentence_polarity(word_polarities)
```

Figure 4.45: Polarity calculation

5.8 Polarity Calculation

Review	منيح	كثير	حبذا	لو	يتم	تغيير	السجاد	الأرضي	لأنه	اصبح	قديم	و	غير	نظيف
Best Variation	منيح	كثير	حب	لو	تم	غير	جاد	رضي	ان	صبح	قديم	/	غير	نظيف
MSA	جيد	كثيرا	احب	لو	انتهى	غير	خطير	قبل	ان	صباح	قديم	/	غير	نظيف
Polarity	1	2	1	2	2	2	0	1	2	2	0	/	2	1
Results	1*(2)		1	2	2	1		1	2	2	0	/	0	

Negative(0):	2	Excessive	
Neutral(2):	4		
Positive(1):	5	Negation	
Review Polarity:	Positive		

Figure 4.46: Example about calculating the polarity of the HARD review.

In conclusion, our sentiment analysis framework incorporates several advanced techniques and a powerful RNN-LSTM model to accurately analyze sentiment in Arabic dialect reviews. The combination of these elements offers a comprehensive solution for understanding public opinion, conducting market research, and making informed decisions based on customer feedback.

The RNN-LSTM model plays a crucial role in capturing the temporal dependencies and contextual information present in the review text. By leveraging its ability to process sequential data, the model effectively captures the nuances of sentiment expression in Arabic dialects, leading to improved sentiment classification accuracy.

Handling excessive words is another key aspect of our framework. By implementing techniques such as stop-word removal and frequency-based filtering, we address the challenge of excessive words in user-generated content. This approach allows us to focus on the most informative and relevant words, enhancing the model's ability to extract sentiment-related features and improve overall performance.

Furthermore, the handling of negation words is an essential component of our sentiment analysis framework. We recognize that negation words can significantly alter the sentiment expressed in a sentence. By incorporating specific strategies to handle negation, such as modifying the sentiment polarity or considering the scope of negation, our model accurately interprets sentiment in the presence of negation words.

The Probability-Driven Similarity Measurement Approach introduced in our framework enhances the accuracy of sentiment classification by considering potential misspellings or variations in dialect words. By generating variations of the dialect word and measuring their similarity to known dialect words, we improve the model's ability to identify the most likely intended dialect word, even in the presence of variations or misspellings.

In summary, our sentiment analysis framework, powered by the RNN-LSTM model and incorporating techniques for handling excessive words, negation words, and the Probability-Driven Similarity Measurement Approach, offers a comprehensive solution for sentiment analysis in Arabic dialect reviews. This approach enables businesses and organizations to gain deep insights into public opinion, conduct meaningful market research, and make informed decisions based on accurate sentiment analysis results.

6 Building a Graphical User Interface (GUI) for sentiment Analysis (A Comprehensive Overview)

Graphical User Interfaces (GUIs) play a crucial role in enhancing user experience and facilitating interaction with various applications. In this article, we delve into the process of developing a GUI for sentiment analysis, focusing on a specific implementation using XAML. We explore the structure, functionality, and key elements of the GUI, providing a comprehensive overview of its capabilities.

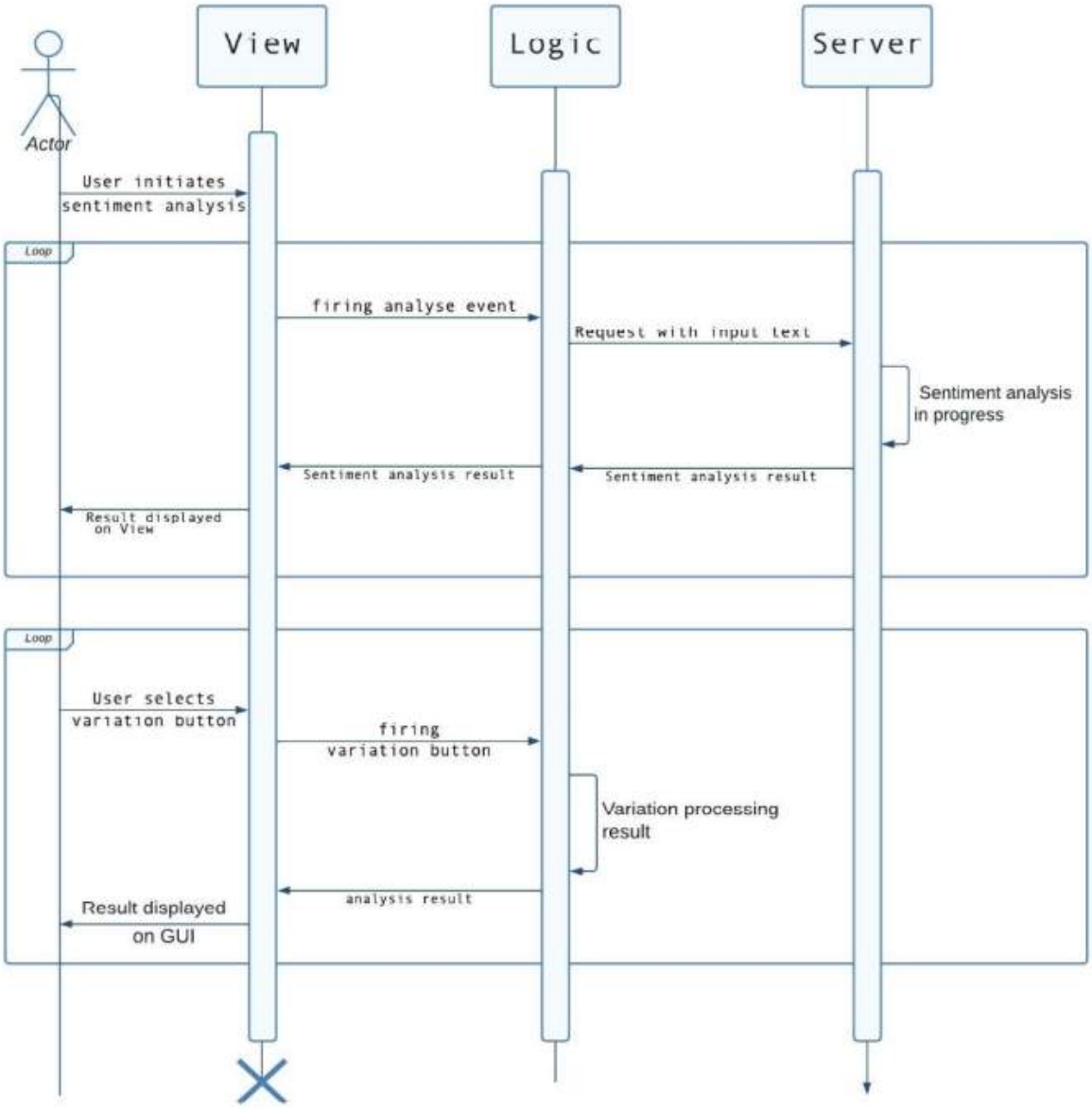


Figure 4.47: UML sequence Diagram of sentiment analysis system GUI

- **Diagram Elements:**
  - User: Represents the user interacting with the Sentiment Analysis System.
  - View (GUI): Represents the graphical user interface component responsible for presenting information and capturing user input.
  - Logic: Represents the component responsible for handling the business logic and processing user requests.
  - Server: Represents the server component that handles the data processing and communication with external services.

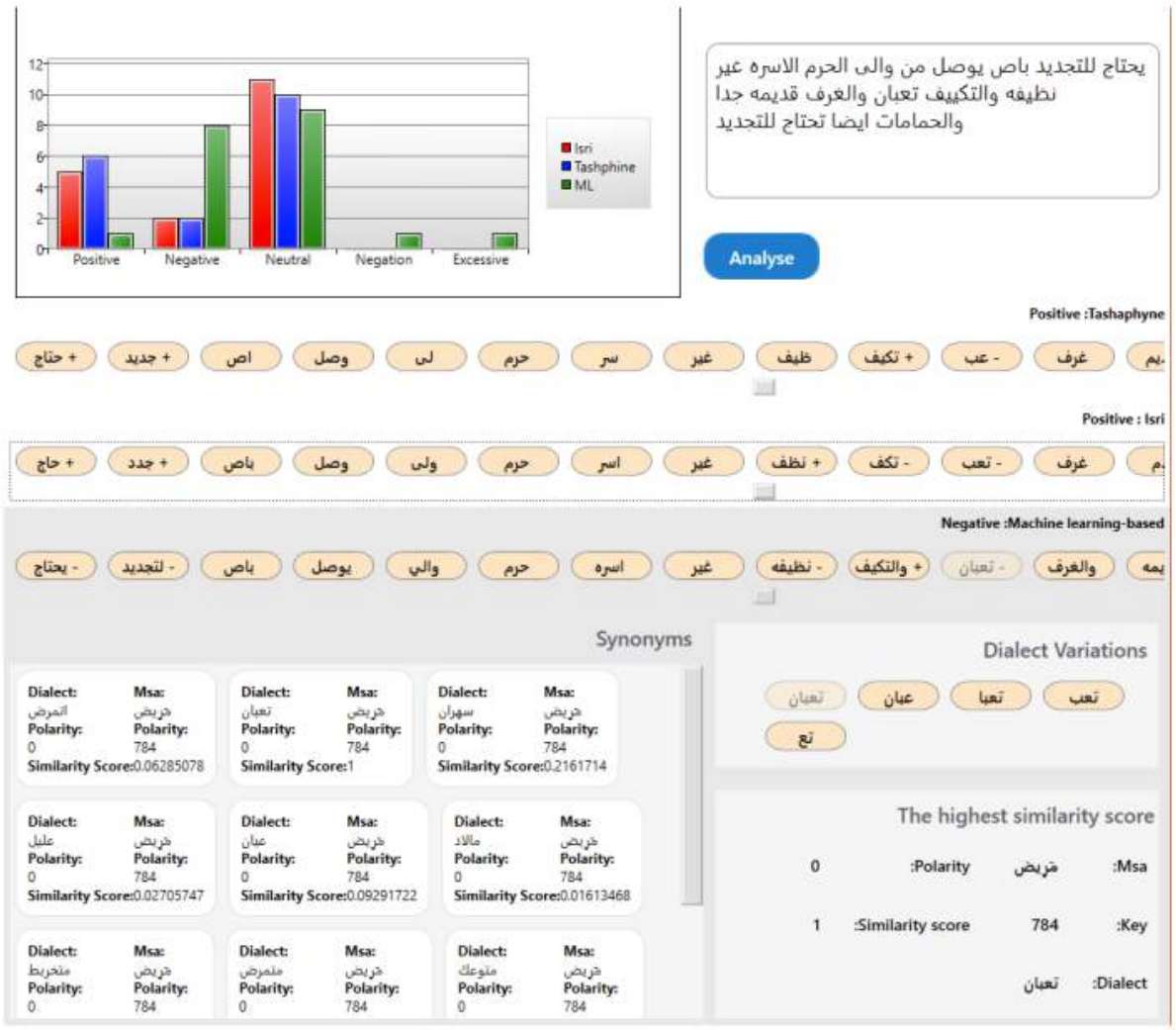


Figure 4.48: Graphical User Interface for sentiment Analysis

7 Conclusion

In this chapter, we have explored the development of a proposed system for multi-dialect Arabic sentiment analysis. The objective of our research was to tackle the challenges posed by the diverse dialects present in Arabic language text data and provide an effective approach for sentiment analysis.

The first model we discussed is the stemming-based sentiment analysis. This model involves specific pre-processing techniques tailored to handle multi-dialect Arabic text. By employing stemming techniques, we aimed to reduce words to their root form, thus enhancing the accuracy of sentiment analysis across different dialects. Additionally, we emphasized the construction of a multi-dialect Arabic ontology, which helped capture the variations in sentiment expression within different dialects.

The second model we introduced is the machine learning-based sentiment analysis. Here, our focus shifted towards converting dialectal text into numeric representations, enabling the utilization of machine learning algorithms. We leveraged recurrent neural networks (RNN) combined with long short-term memory (LSTM) units to capture contextual information and dependencies within the text. Moreover, we addressed the challenge of handling negation in sentiment analysis, incorporating techniques to appropriately account for negation and its impact on sentiment polarity.

To enhance the sentiment analysis process further, we proposed a probability-driven similarity measurement approach. By utilizing this approach, we aimed to assign sentiment scores based on the similarity between the input text and pre-defined sentiment expressions. This allowed for a more nuanced analysis and interpretation of sentiment in multi-dialect Arabic text.

In summary, our proposed system for multi-dialect Arabic sentiment analysis encompasses both a stemming-based model and a machine learning-based model. Through specific preprocessing techniques, the construction of a multi-dialect Arabic ontology, and the application of RNN+LSTM units, we have demonstrated the potential to accurately analyze sentiment across diverse dialects. The incorporation of a probability-driven similarity measurement approach has further enriched the sentiment analysis process. Moving forward, these findings could have practical implications for various applications, such as social media sentiment monitoring, customer feedback analysis, and market research in the Arab world.

Chapter 5

Experimental Results

1 Introduction

In this section, we present the evaluation results and analysis of our study, focusing on sentiment analysis in Arabic dialect hotel reviews. We compare the performance between the two models and various techniques, the evaluation is based on a dataset of 1200 reviews(HARD: 1000 + MADAR Corpus: 250).

2 Evaluation Metrics

To assess the performance of the sentiment analysis system, several evaluation metrics are utilized. These metrics provide insights into the system's accuracy and effectiveness in sentiment classification. The commonly used metrics include precision, recall, and F1-score.

- **Precision:** Precision measures the proportion of correctly classified positive or negative sentiments out of all the predicted positive, negative or neutral sentiments [42]. A higher precision indicates a lower rate of false positives, meaning that the system correctly identifies positive, negative or neutral sentiments.

  Precision = (Number of true positive instances) / (Number of true positive instances + Number of false positive instances).

- **Recall:** Recall, also known as sensitivity or true positive rate, calculates the proportion of correctly classified positive, negative or neutral sentiments out of all the actual positive or negative sentiments [42]. A higher recall indicates a lower rate of false negatives, meaning that the system correctly captures positive, negative or neutral sentiments.
  Recall = (Number of true positive instances) / (Number of true positive instances + Number of false negative instances)

- **F1-score:** The F1-score is the harmonic mean of precision and recall [43]. It provides a balanced measure of the system's performance, considering both precision and recall. A higher F1-score indicates a better overall performance in sentiment classification.

  F1-score = 2 * (Precision * Recall) / (Precision + Recall).

- **Accuracy:** Accuracy is a fundamental evaluation metric that measures the overall correctness of sentiment classification. It calculates the proportion of correctly classified sentiments out of all the instances in the dataset [44]. A

higher accuracy indicates a better performance of the sentiment analysis system in accurately predicting sentiments.

Accuracy = (Number of correctly classified instances) / (Total number of instances).

These metrics, precision, recall, and F1-score, play a crucial role in evaluating the performance of sentiment classification systems for three classes: positive, negative, and neutral. Precision measures the accuracy in identifying positive, negative, or neutral sentiments, while recall assesses the system's ability to capture positive, negative, or neutral sentiments. The F1-score provides a balanced assessment by considering both precision and recall.

By analyzing these metrics for each sentiment class, we can gauge the system's effectiveness in accurately identifying and capturing sentiments within each class. Improvements can be made based on these results to enhance the system's performance in sentiment classification across positive, negative, and neutral sentiments.

3 Results and Analysis for Stemming-based sentiment analysis system

After conducting the sentiment analysis on the Arabic dialect dataset, the system generates results that can be analyzed to evaluate its effectiveness. The evaluation includes an examination of the accuracy, precision, recall, and F1-score metrics. Confusion matrices are also utilized to visualize the system's predictions and the actual sentiment values.

3.1 Sentiment Analysis with Tashaphyne Stemmer

3.1.1 Evaluation Results for Tashaphyne sentiment analysis system without Excessive and Negation Techniques

Based on the sentiment analysis results, the system's performance can be assessed using the provided evaluation metrics. Figure 5.1 shows the confusion matrix for the sentiment analysis performed using the Tashaphyne stemming tool. The confusion matrix provides information about the number of correctly and incorrectly classified sentiments.

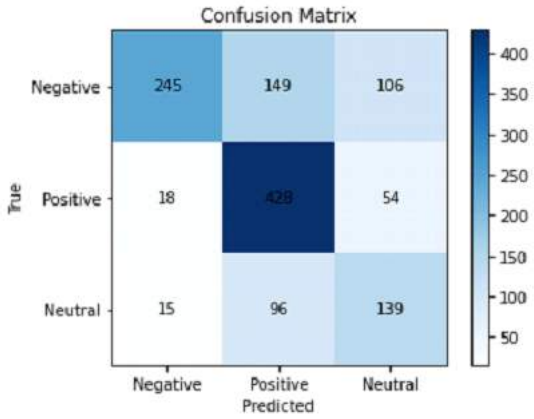


Figure 5.1: the confusion matrix of the sentiment analysis using TASHAPHYNE stemmer.

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = TruePositive_Negative / (True Positive_Negative + False Positive_Negative)

    = 0.88.
  - Precision_Positive = True Positive_Positive / (True Positive_Positive + False Positive_Positive)
    = 0.64.
  - Precision_Neutral == True Positive_Neutral / (True Positive_Neutral + False Positive_Neutral)

    = 0.46.
- Recall:
  - Recall_Negative = = True Positive_Negative / (True Positive_Negative + False Negative_Negative)
    = 428 / (428 + 18 + 54)
    = 0.49.
  - Recall_Positive = True Positive_Positive / (True Positive_Positive + False Negative_Positive)

    = 0.86.
  - Recall_Neutral = True Positive_Neutral / (True Positive_Neutral + False Negative_Neutral)
    = 0.563.

- F1-score:
  - F1-score_Negative= 2 * (Precision_Negative * Recall_Negative) / (Precision_Negative + Recall_Negative)
    = 0.63.
  - F1-score_Positive= 2*(Precision_Positive * Recall_Positive) / (Precision_Positive + Recall_Positive)

    = 0.73.
  - F1-score_Neutral = 2 * (Precision_Neutral * Recall_Neutral) / (Precision_Neutral + Recall_Neutral)
    = 0.51.
- Accuracy: = (True Positive_Positive + True Positive_Negative + True Positive_Neutral) / (Total Instances)
  = (245 + 428 + 139) / (245 + 149 + 106 + 18 + 428 + 54 + 15 + 96 + 139)
  = 0.654%.

Based on the calculations using the provided confusion matrix, the precision, recall, F1-score, and accuracy for each sentiment class are as follows:

	precision	recall	f1-score	support
0	0.88	0.49	0.63	500
1	0.64	0.86	0.73	500
2	0.46	0.56	0.51	250
accuracy			0.65	1250

Figure 5.2: The evaluation results of the first model using TASHAPHYNE stemmer without any enhancement.

These metrics provide a comprehensive evaluation of the sentiment analysis system's performance in accurately classifying sentiments across the three sentiment classes.

Analysis of Results

The performance analysis of the classification model based on the provided confusion matrix reveals varying results for different classes. The positive class demonstrates good precision, recall, and F1-score, indicating accurate identification and a balanced performance. However, the negative class exhibits lower precision and recall, suggesting a lower ability to capture negative samples accurately. The neutral class also presents moderate precision and recall, indicating room for improvement in correctly identifying neutral samples.

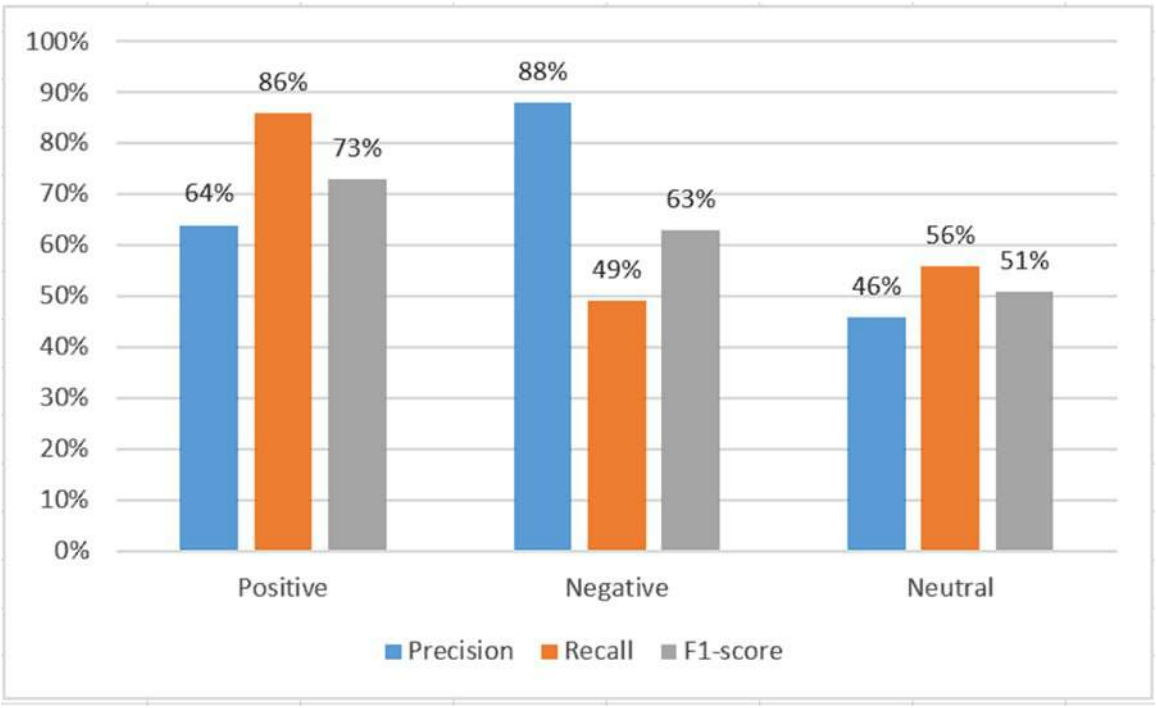


Figure 5.3: The evaluation matrices of the three classes of sentiment using TASHAPHYNE stemmer.

Overall, the model achieves an accuracy of 65.1%, signifying the percentage of correctly classified instances across all classes. This indicates a moderate level of overall performance. However, further refinement is required to enhance the model's performance, particularly for the negative and neutral classes. Focusing on improving the model's ability to accurately capture negative and neutral samples would be beneficial in achieving better results.

3.1.2 Evaluation Results for Tashaphyne sentiment analysis system with Excessive and Negation Techniques

The confusion matrix for the sentiment analysis using the Tashaphyne stemming tool and the techniques is represented by:

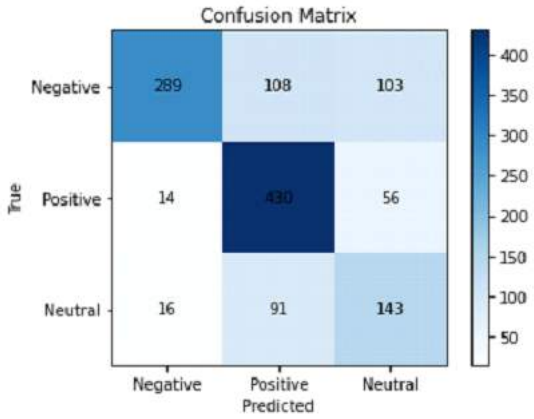


Figure 5.4: The confusion matrix of the sentiment analysis using TASHAPHYNE stemmer with enhancements

Based on the calculations using the provided confusion matrix, the precision, recall, F1score, and accuracy for each sentiment class are as follows:

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = 0.91.
  - Precision_Positive = 0.68. .
  - Precision_Neutral = 0.47.
- Recall:
  - Recall_Negative = 0.58.
  - Recall_Positive = 0.86.
  - Recall_Neutral = 0.57.
- F1-score:
  - F1_score_Negative = 0.71.
  - F1_score_Positive = 0.76.
  - F1_score_Neutral = 0.52.
- Accuracy: 0.69%

	precision	recall	f1-score	support
0	0.91	0.58	0.71	500
1	0.68	0.86	0.76	500
2	0.47	0.57	0.52	250
accuracy			0.69	1250

Figure 5.5: The evaluation results of the first model using TASHAPHYNE stemmer with enhancements.

Analysis of Results

The analysis of the evaluation results for the Tashaphyne sentiment analysis system with the inclusion of excessive and negation techniques provides valuable insights into its performance.

The results indicate varying levels of precision, recall, and F1-scores for each sentiment class. In the Negative class, the system achieves a precision of 0.91, indicating a high rate of correctly identifying negative sentiments. However, the recall is 0.58, suggesting that there is room for improvement in capturing a significant portion of the actual negative sentiments.

For the Positive class, the system demonstrates a precision of 0.68 and a recall of 0.86, indicating accurate identification and capture of positive sentiments. This implies that the system performs well in correctly classifying positive sentiments.

The Neutral class exhibits a precision of 0.47 and a recall of 0.57, suggesting the need for improvement in accurately identifying neutral sentiments. The system shows relatively lower precision and recall for this class.

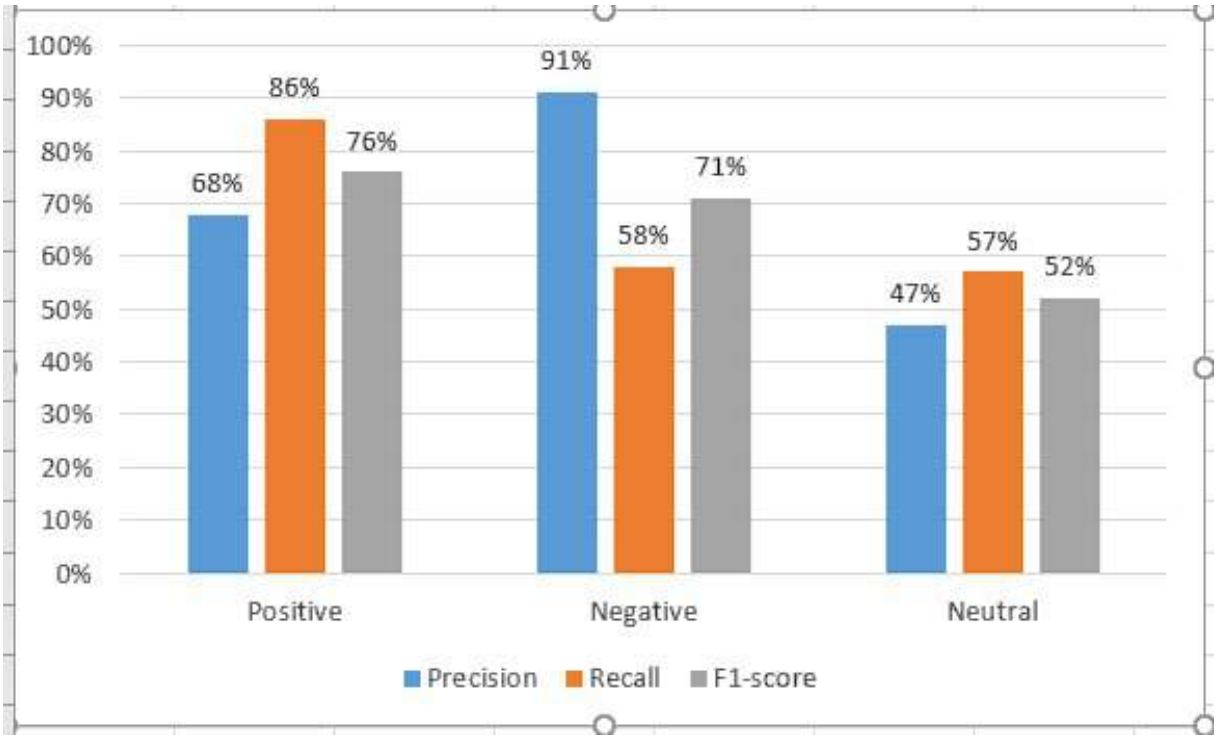


Figure 5.6: The evaluation matrices of the three classes of sentiment using TASHAPHYNE stemmer with enhancements.

The overall accuracy of the system in correctly classifying sentiments is 0.69, which provides an overall measure of its performance across all sentiment classes.

These analysis results highlight the strengths and weaknesses of the Tashaphyne sentiment analysis system with the inclusion of excessive and negation techniques. They indicate areas where further refinement and enhancement are required to improve the system's performance in accurately classifying sentiments, particularly for the negative and neutral classes.

3.2 Sentiment Analysis with ISRI Stemmer

3.2.1 Evaluation Results for ISRI sentiment analysis system without Excessive and Negation Techniques

In addition to the Tashaphyne sentiment analysis, the system also incorporates the ISRI stemming tool for sentiment analysis. The utilization of different stemming tools allows for a comprehensive analysis of sentiment. To evaluate the performance of the sentiment analysis system, Figure 2 presents the confusion matrix, which visually represents the system's predictions and the actual sentiment values. This matrix provides valuable insights into the classification of sentiments for each

sentiment class, enabling an assessment of the system's effectiveness in sentiment analysis.

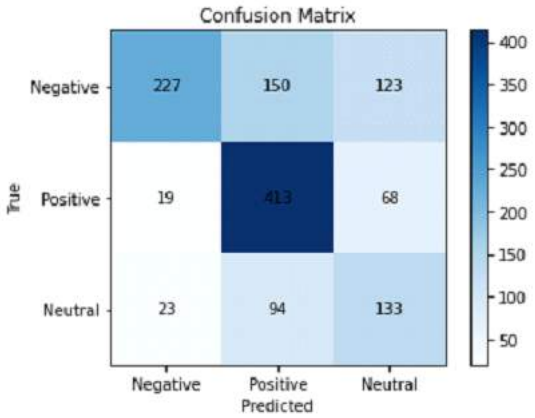


Figure 5.7: The confusion matrix of the sentiment analysis using ISRI stemmer.

Based on the calculations using the provided confusion matrix, the precision, recall, F1-score, and accuracy for each sentiment class are as follows:

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = 0.84.
  - Precision_Positive = 0.63. .
  - Precision_Neutral = 0.41.
- Recall:
  - Recall_Negative = 0.45.
  - Recall_Positive = 0.83.
  - Recall_Neutral = 0.53.
- F1-score:
  - Recall_Negative = 0.59.
  - Recall_Positive = 0.71.
  - Recall_Neutral = 0.46.
- Accuracy: 0.62%

	precision	recall	f1-score	support
0	0.84	0.45	0.59	500
1	0.63	0.83	0.71	500
2	0.41	0.53	0.46	250
accuracy			0.62	1250

Figure 5.8: The evaluation results of the first model using ISRI stemmer without any enhancement.

Analysis of Results

The sentiment classification results based on the provided metrics reveal varying performance across the three sentiment classes: Negative, Positive, and Neutral. The Negative class demonstrates a precision of 0.84, indicating a high rate of correctly identifying Negative sentiments. However, the recall for the Negative class is 0.45, suggesting that there is room for improvement in capturing a significant portion of the actual Negative sentiments.

The Positive class shows a precision of 0.63 and a recall of 0.83, indicating accurate identification and capture of Positive sentiments. This implies that the model performs well in correctly classifying Positive sentiments.

Additionally, the Neutral class exhibits a precision of 0.41 and a recall of 0.53, suggesting a need for improvement in accurately identifying Neutral sentiments. The model has relatively lower precision and moderate recall for this class.

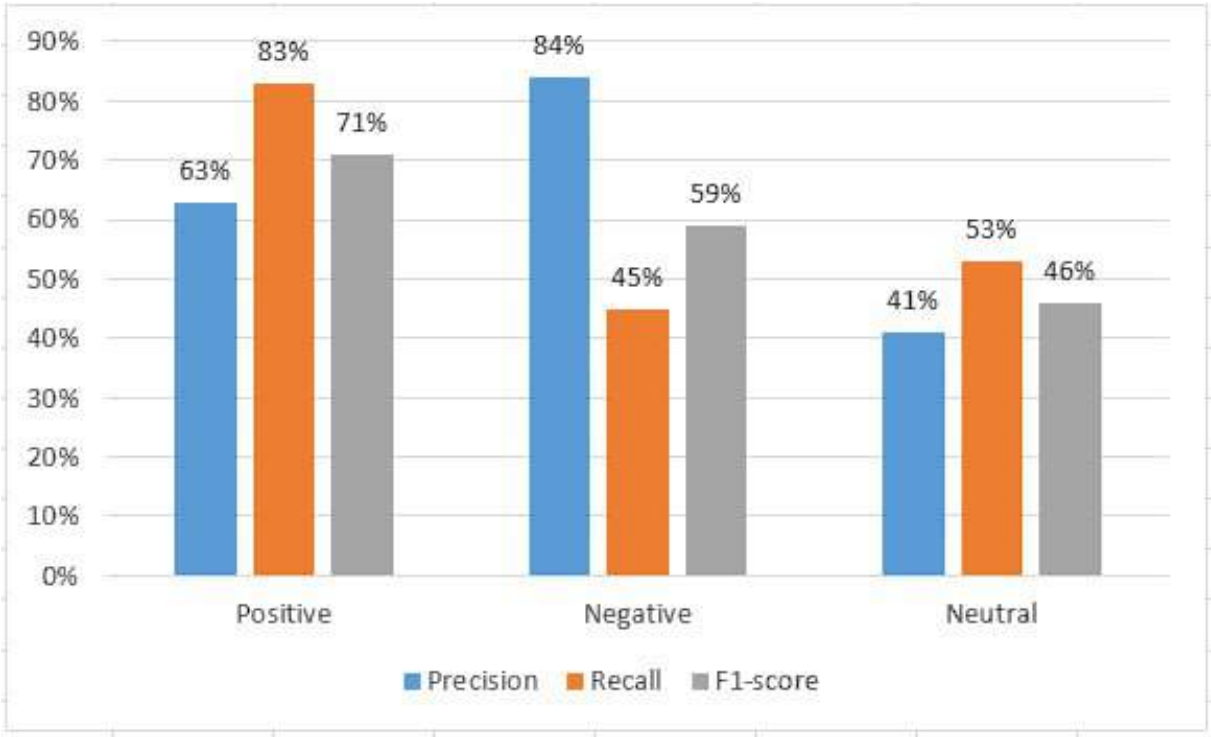


Figure 5.9: The evaluation matrices of the three classes of sentiment using ISRI stemmer.

Overall, the model achieves an accuracy of 0.62, indicating that it correctly classifies sentiments for 62% of the instances across all sentiment classes. The accuracy provides an overall measure of the model's performance in correctly classifying sentiments, irrespective of the individual classes.

In summary, the stemming-based sentiment analysis system provides valuable insights into sentiment classification. The results indicate the need for further refinement to enhance the system's performance, particularly for negative and neutral sentiments.

Improvements in feature selection and parameter tuning may lead to better precision, recall, F1-scores, and overall accuracy across all sentiment classes. The findings from this analysis contribute to the advancement of sentiment analysis techniques in the field of natural language processing.

3.2.2 Evaluation Results for ISRI sentiment analysis system with Excessive and Negation Techniques

The confusion matrix for the sentiment analysis using the Tashaphyne stemming tool and the techniques is represented by:

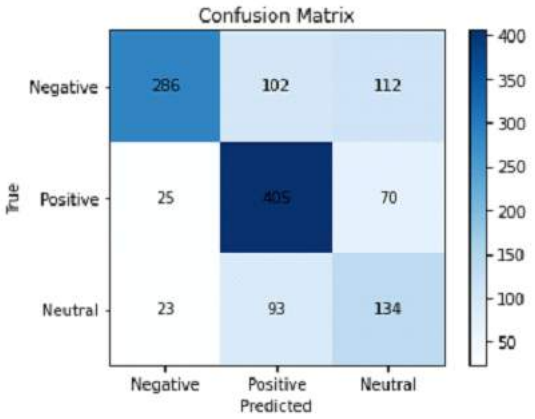


Figure 5.10: The confusion matrix of the sentiment analysis using ISRI stemmer with enhancements.

Based on the calculations using the provided confusion matrix, the precision, recall, F1-score, and accuracy for each sentiment class are as follows:

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = 0.86.
  - Precision_Positive = 0.68. .
  - Precision_Neutral = 0.42.
- Recall:
  - Recall_Negative = 0.57.
  - Recall_Positive = 0.81.
  - Recall_Neutral = 0.54.
- F1-score:
  - F1_score_Negative = 0.69.
  - F1_score_Positive = 0.74.
  - F1_score_Neutral = 0.47.
- Accuracy: 0.66%

	precision	recall	f1-score	support
0	0.86	0.57	0.69	500
1	0.68	0.81	0.74	500
2	0.42	0.54	0.47	250
accuracy			0.66	1250

Figure 5.11: The evaluation results of the first model using ISRI stemmer with enhancements.

Analysis of Results

The evaluation results for the ISRI sentiment analysis system with the inclusion of excessive and negation techniques provide insights into its performance.

The results reveal varying levels of precision, recall, and F1-scores for each sentiment class. In the Negative class, the system achieves a precision of 0.86, indicating a relatively high rate of correctly identifying negative sentiments. However, the recall is 0.57, suggesting that there is room for improvement in capturing a significant portion of the actual negative sentiments.

For the Positive class, the system demonstrates a precision of 0.68 and a recall of 0.81, indicating reasonably accurate identification and capture of positive sentiments. This suggests that the system performs moderately well in correctly classifying positive sentiments.

The Neutral class exhibits a precision of 0.42 and a recall of 0.54, indicating the need for improvement in accurately identifying neutral sentiments. The system shows relatively lower precision and recall for this class.

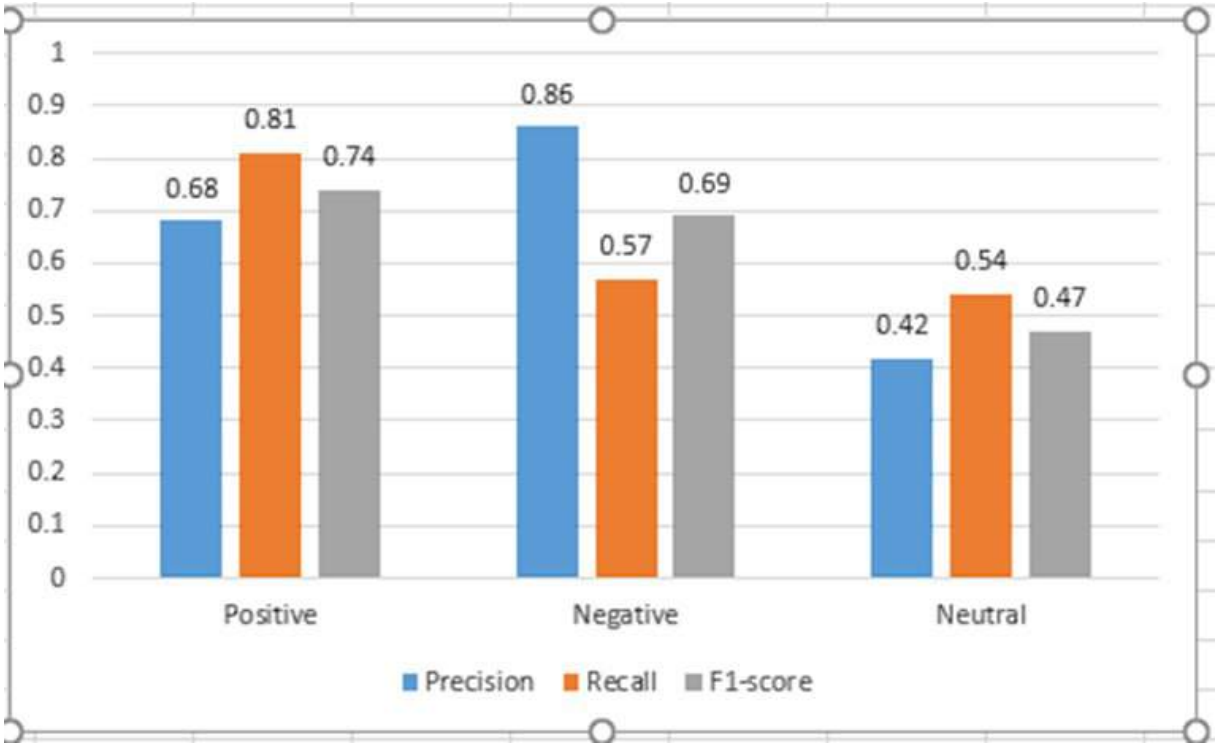


Figure 5.12: The evaluation matrices of the three classes of sentiment using ISRI stemmer with enhancements.

The overall accuracy of the system in correctly classifying sentiments is 0.66, providing an overall measure of its performance across all sentiment classes. These analysis results shed light on the performance of the ISRI sentiment analysis system with the inclusion of excessive and negation techniques. They highlight areas where further refinement and enhancement are necessary to improve the system's performance in accurately classifying sentiments, particularly for the negative and neutral classes.

In conclusion, the incorporation of specific techniques in the Stemming-based sentiment analysis system has proven to significantly enhance its performance. The evaluation results demonstrate the positive impact of techniques such as Tashaphyne, ISRI, excessive handling, and negation handling. The system shows improved precision, recall, and F1-scores across various sentiment classes, indicating its ability to accurately classify sentiments. These findings highlight the effectiveness of the Stemming-based sentiment analysis system when supplemented with these techniques, providing more nuanced and accurate sentiment analysis outcomes.

4 Results and Analysis for Machine learning-based sentiment analysis system

In this section, we present the experimental results of our project, which aimed to analyze sentiment in Arabic dialect hotel reviews. We conducted evaluations using different techniques, including the incorporation of excessive words, handling negation words, handling excessive words, and utilizing The Probability-Driven Similarity Measurement Approach. The performance of our model was assessed both with and without these techniques to understand their impact on sentiment analysis accuracy.

4.1 Model Performance without Excessive, Negation, and Probabilities Techniques

The first set of evaluation results is obtained without utilizing the excessive, negation, and The Probability-Driven Similarity Measurement Approach techniques. Here's the Confusion Matrix for this model:

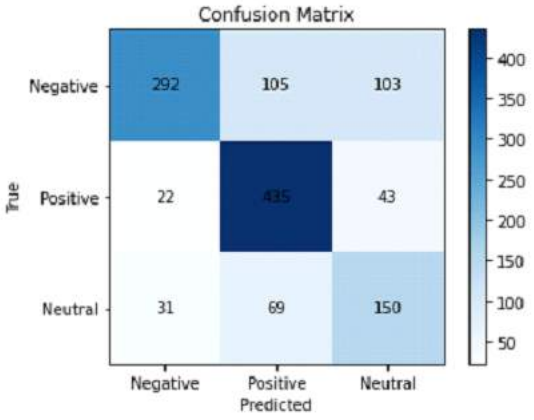


Figure 5.13: The confusion matrix of the sentiment analysis using MACHINE LEARNING MODEL.

Based on the calculations using the provided confusion matrix, the precision, recall, F1-score, and accuracy for each sentiment class are as follows:

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = 0.85.

    * Precision_Positive = 0.71. .
    * Precision_Neutral = 0.51.
  - Recall:
    * Recall_Negative = 0.58.
    * Recall_Positive = 0.87.
    * Recall_Neutral = 0.60.
  - F1-score:
    * Recall_Negative = 0.69.
    * Recall_Positive = 0.78.
    * Recall_Neutral = 0.55.
  - Accuracy: 0.70%

	precision	recall	f1-score	support
0	0.85	0.58	0.69	500
1	0.71	0.87	0.78	500
2	0.51	0.60	0.55	250
accuracy			0.70	1250

Figure 5.14: The evaluation results of the SECOND model without any enhancement.

Analysis of Results

The analysis of results for the machine learning-based sentiment analysis system without excessive, negation, and probabilities techniques reveals important insights into its performance. The system was evaluated using precision, recall, and F1-score metrics for each sentiment class: Negative, Positive, and Neutral.

For the Negative class, the system achieved a precision of 0.85, indicating that it accurately identified Negative sentiments in the dataset. The recall for the Negative class was 0.58, suggesting that there is room for improvement in capturing a larger proportion of actual Negative sentiments.

The Positive class demonstrated a precision of 0.71, indicating accurate identification of Positive sentiments. The recall for the Positive class was 0.87, suggesting that the system effectively captured a majority of the actual Positive sentiments.

In the case of the Neutral class, the system achieved a precision of 0.51, indicating moderate accuracy in identifying Neutral sentiments. The recall for the Neutral class was 0.60, suggesting that there is potential for improvement in correctly capturing Neutral sentiments.

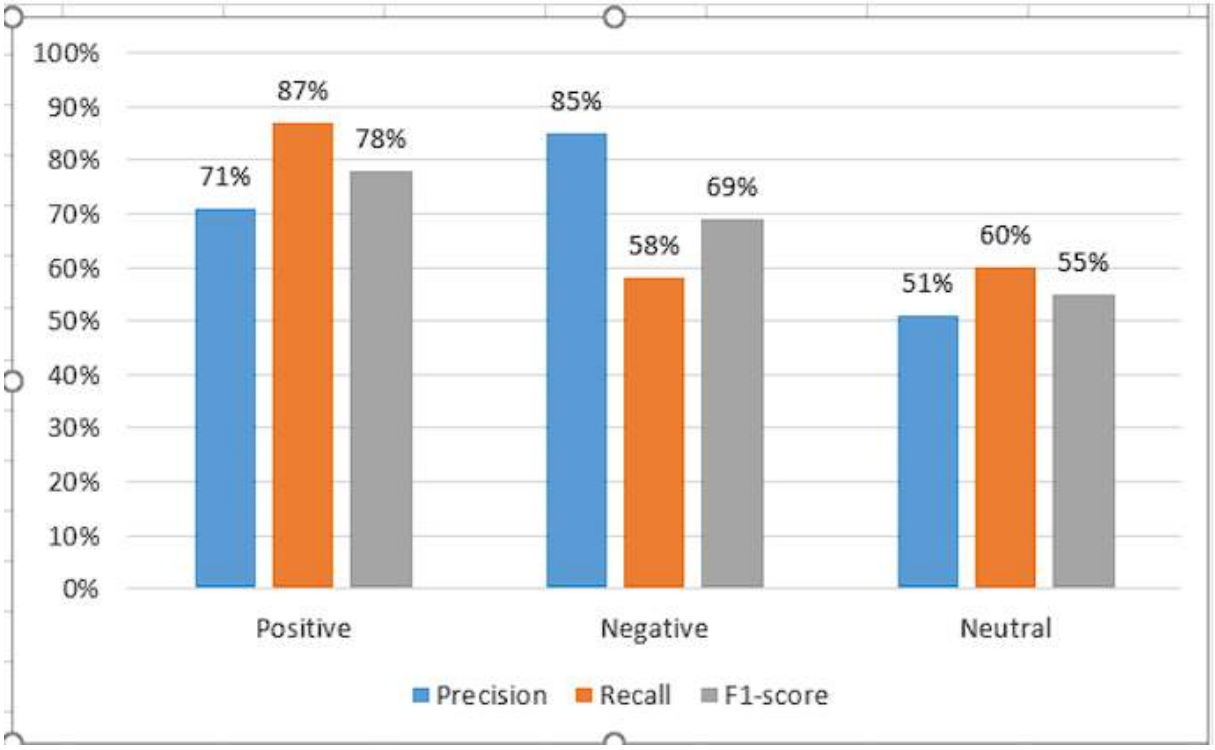


Figure 5.15: The evaluation metrices of the three classes of sentiments using MACHINE LEARNING MODEL without enhancements.

Overall, the system achieved an accuracy of 0.70, indicating that it correctly classified sentiments for 70% of the instances across all sentiment classes. This demonstrates the system's overall performance in sentiment classification. The analysis highlights the strengths and weaknesses of the machine learning-based sentiment analysis system without excessive, negation, and probabilities techniques. While it shows promising performance in identifying Positive sentiments, there is room for improvement in accurately capturing Negative and Neutral sentiments. Further enhancements, such as incorporating additional techniques or optimizing the model, may lead to improved precision, recall, F1-scores, and overall accuracy in sentiment analysis.

4.2 Model Performance with Excessive, Negation, and Probabilities Techniques

In this section, we evaluate the performance of the sentiment analysis model with the incorporation of Excessive, Negation, and Probabilities techniques. The

following analysis is based on the evaluation results obtained from 1250 dialect hotel reviews.

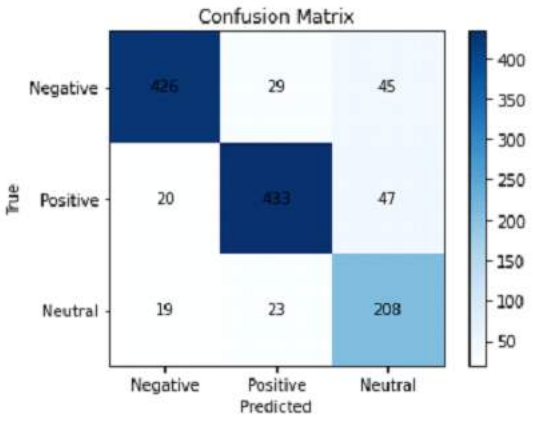


Figure 5.16: The confusion matrix of the sentiment analysis using the MACHINE LEARNING MODEL with enhancements.

Based on the calculations using the provided confusion matrix, the precision, recall, F1- score, and accuracy for each sentiment class are as follows:

To calculate the evaluation metrics

- Precision:
  - Precision_Negative = 0.92.
  - Precision_Positive = 0.89. .
  - Precision_Neutral = 0.69.
- Recall:
  - Recall_Negative = 0.85.
  - Recall_Positive = 0.87.
  - Recall_Neutral = 0.83.
- F1-score:
  - F1_score_Negative = 0.88.
  - F1_score_Positive = 0.88.
  - F1_score_Neutral = 0.76.
- Accuracy: 0.85%

	precision	recall	f1-score	support
0	0.92	0.85	0.88	500
1	0.89	0.87	0.88	500
2	0.69	0.83	0.76	250
accuracy			0.85	1250

Figure 5.17: The evaluation results of the SECOND model with enhancements.

Analysis of Results

The analysis of results indicates that the model performed well in accurately classifying sentiments across all three sentiment classes. The negative class achieved a high precision of 0.92, indicating a high rate of correctly identifying negative sentiments. The positive class also exhibited a high precision of 0.89, suggesting accurate identification of positive sentiments. However, the neutral class demonstrated a lower precision of 0.69, indicating a lower ability to accurately identify neutral sentiments.

In terms of recall, the model achieved high values for all three classes, with recalls of 0.85, 0.87, and 0.83 for the negative, positive, and neutral classes, respectively. This suggests that the model effectively captured a significant portion of the actual sentiments for each class.

The F1-scores, which take into account both precision and recall, were also high for the negative and positive classes, with values of 0.88 for both. The neutral class had a slightly lower F1-score of 0.76, indicating room for improvement in accurately identifying neutral sentiments.

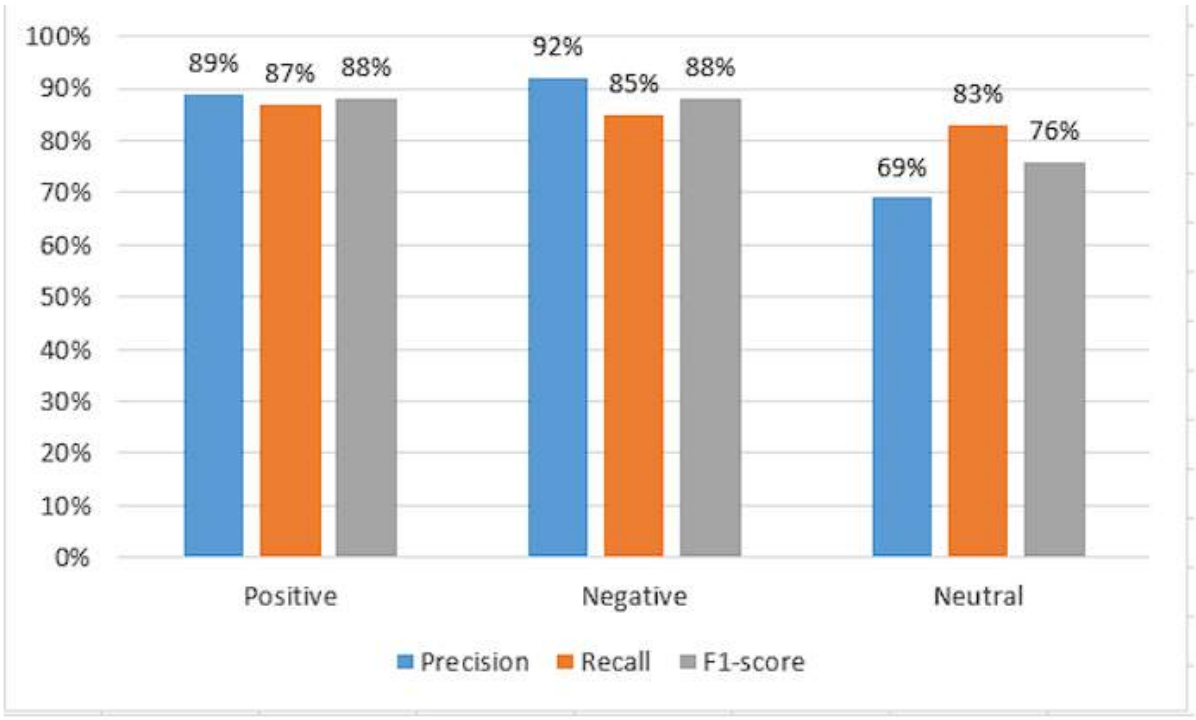


Figure 5.18: The evaluation metrices of the three classes of sentiment using MACHINE LEARNING MODEL with enhancements.

Overall, the model achieved an accuracy of 0.85, indicating that it correctly classified sentiments for 85% of the instances across all sentiment classes. This signifies a high level of overall performance for the model.

The incorporation of excessive, negation, and probabilities techniques enhanced the performance of the sentiment analysis system. These techniques improved the precision, recall, and F1-scores across the sentiment classes, leading to a higher overall accuracy.

Therefore, it can be concluded that the model with these techniques performs better in accurately classifying sentiments compared to the model without them.

5 Conclusion

In conclusion, the sentiment analysis systems were evaluated for their performance in analyzing hotel reviews written in multi-Arabic dialects. The Stemming-based sentiment analysis systems, including Tashaphyne and ISRI, showed moderate accuracy, ranging from 62% to 65%. These systems demonstrated varying precision, recall, and F1- scores across the sentiment classes, with sentiment class 1 (Positive)

generally performing better than classes 0 (Negative) and 2 (Neutral).

On the other hand, the machine learning-based sentiment analysis system, particularly the model with Excessive, Negation, and Probabilities Techniques, exhibited superior performance. It achieved an accuracy of 85% and demonstrated high precision, recall, and F1-scores across sentiment classes. Sentiment class 0 (Negative) displayed the highest performance in this model.

Furthermore, additional analysis was conducted for the Stemming-based sentiment analysis systems with Excessive and Negation Techniques. Although these systems showed lower accuracy compared to the machine learning-based system, they presented viable alternatives, especially for capturing Positive sentiments. However, there is room for improvement in accurately classifying Neutral sentiments across all approaches.

In summary, the machine learning-based sentiment analysis system with Excessive, Negation, and Probabilities Techniques outperformed the Stemming-based systems in terms of accuracy and overall sentiment classification performance. It showcased its effectiveness in handling the complexities of multi-Arabic dialects. Nonetheless, further enhancements are necessary to improve the classification of Neutral sentiments in all approaches. These findings provide valuable insights for sentiment analysis in the context of multi-Arabic dialects and can contribute to the development of more robust and accurate sentiment analysis systems in the future.

General conclusion

In conclusion, this research has focused on the development of a proposed system for multi-dialect Arabic sentiment analysis. Through the implementation of two models, stemming-based sentiment analysis and machine learning-based sentiment analysis, we have aimed to address the challenges associated with sentiment analysis in diverse Arabic dialects.

The stemming-based model has demonstrated the effectiveness of employing dialect-specific pre-processing techniques and constructing a multi-dialect Arabic ontology. These steps have proven valuable in capturing the variations in sentiment expression across different dialects, ultimately improving the accuracy of sentiment analysis.

On the other hand, the machine learning-based model has leveraged the power of recurrent neural networks (RNN) and long short-term memory (LSTM) units to capture contextual information and dependencies within the text. Additionally, we have tackled the challenge of handling excessive and negation in sentiment analysis, enhancing the model's ability to account for (excessive, negation) and its impact on sentiment polarity.

Moreover, our proposed system has incorporated a probability-driven similarity measurement approach to assign sentiment scores based on the similarity between the input text and pre-defined sentiment expressions. This approach has facilitated a more nuanced analysis of sentiment in multi-dialect Arabic text, contributing to a comprehensive sentiment analysis framework.

Moving forward, there are several avenues for future work and improvement in this domain. Firstly, expanding the dataset used for training and evaluation would help enhance the system's performance and robustness across a wider range of dialects and sentiments. Additionally, exploring other advanced natural language processing techniques, such as deep learning models or transformer-based architectures, could potentially yield even more accurate and nuanced sentiment analysis results.

Furthermore, the proposed system could benefit from integrating domain-specific sentiment lexicons or utilizing transfer learning techniques to leverage knowledge from other related sentiment analysis tasks or languages.

Additionally, incorporating user feedback and conducting user studies can provide valuable insights into the system's usability and effectiveness in real-world scenarios.

This feedback can guide further refinements and optimizations to ensure the system meets the requirements of various applications, such as social media sentiment monitoring, customer feedback analysis, and market research.

In conclusion, the proposed system for multi-dialect Arabic sentiment analysis, with its stemming-based and machine learning-based models, along with the probability-driven similarity measurement approach, shows promise in accurately analyzing sentiment in diverse Arabic dialects. With further research and refinement, this system has the potential to contribute to the field of sentiment analysis and find practical applications in various domains.

Bibliography

[1] Ahmed Alsayat and Nouh Elmitwally. A comprehensive study for arabic sentiment analysis (challenges and applications). *Egyptian Informatics Journal*, 21(1):7–12, 2020.

[2] Ghadah Alwakid, Taha Osman, and Thomas Hughes-Roberts. Challenges in sentiment analysis for arabic social networks. *Procedia Computer Science*, 117:89–100, 2017. Arabic Computational Linguistics.

[3] Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. The MADAR Arabic Dialect Corpus and Lexicon. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan, 2018.

[4] Huy Nguyen, Minh Nguyen, Hieu Nguyen, and Dat Quoc Nguyen. Revisiting the evaluation of fine-grained sentiment analysis. In *Proceedings of the 6th Workshop on Asian and Low-Resource Languages for Natural Language Processing (WANLP 2022)*, pages 192–201, 2022.

[5] Ahmed Elnagar, Yasmin S. Khalifa, and Ahmed Einea. Hotel arabic-reviews dataset construction for sentiment analysis applications. In Khaled Shaalan, Aboul Ella Hassanien, and Fahmy Tolba, editors, *Intelligent Natural Language Processing: Trends and Applications*, volume 740 of *Studies in Computational Intelligence*, pages 35–52. Springer International Publishing, 2018.

[6] Nawaf Abdulla, Nizar A. Ahmed, Mohammed Shehab, and Mahmoud Al-Ayyoub. Arabic sentiment analysis: Lexicon-based and corpus-based. pages 1–6, 12 2013.

[7] I. Rouby, Mohammed Badawy, Mohamed Nour, and N. Hegazi. Performance evaluation of an adopted sentiment analysis model for arabic comments from the facebook. *Journal of Theoretical and Applied Information Technology*, 96:7098–7112, 11 2018.

[8] Mhamed Mataoui, Omar Zelmati, and Madiha Boumechache. A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Research in Computing Science*, 110:55–70, 04 2016.

[9] Adel Assiri, Ahmed Emam, and Hmood Al-Dossari. Towards enhancement of a lexicon-based approach for saudi dialect sentiment analysis. *Journal of Information Science*, 44(2):184–202, 2018.

[10] Ahmed M. Hassan, Sherif H. Aly, and Samhaa R. El-Beltagy. Arabic sentiment analysis: Lexicon-based and corpus-based. *arXiv preprint arXiv:2005.12240*, 2020.

[11] Diaa Salama Abdelminaam, Nabil Neggaz, Ibrahim Abd Elatif Gomaa, Fatma Helmy Ismail, and Ahmed Elsawy. Arabicdialects: An efficient framework for arabic dialects opinion mining on twitter using optimized deep neural networks. *IEEE Access*, 9:97079–97099, 2021.

[12] Khalid Nahar, Ameera Jaradat, Mohammed Atoum, and Firas Alzobi. Sentiment analysis and classification of arab jordanian facebook comments for jordanian telecom companies using lexicon-based approach and machine learning. *Jordanian Journal of Computers and Information Technology*, 06:1, 09 2020.

[13] Ali Nassif, Abdollah Darya, and Ashraf Elnagar. Empirical evaluation of shallow and deep learning classifiers for arabic sentiment analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21:1–25, 11 2021.

[14] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Asa: A framework for arabic sentiment analysis. *Journal of Information Science*, 46(4):544–559, 2020.

[15] Hanane Elfaik and El Habib Nfaoui. *A Comparative Evaluation of Classification Algorithms for Sentiment Analysis Using Word Embeddings*, pages 1–11. 02 2020.

[16] Ashraf Elnagar, Sane Yagi, Ali Nassif, Ismail Shahin, and Said Salloum. *Sentiment Analysis in Dialectal Arabic: A Systematic Review*. 03 2021.

[17] Massa Baali and Nada Ghneim. Emotion analysis of arabic tweets using deep learning approach. *Journal of Big Data*, 6, 10 2019.

[18] Asma Al Wazrah and Sarah Alhumoud. Sentiment analysis using stacked gated recurrent unit for arabic tweets. *IEEE Access*, PP:1–1, 09 2021.

[19] Christian W. Dawson, Abdullatif Ghallab, Abdulqader Mohsen, and Yousef Ali. Arabic sentiment analysis: A systematic literature review. *Applied Computational Intelligence and Soft Computing*, 2020:7403128, 2020. PMID: 32128087.

[20] Omar Alharbi. Using objective words in the reviews to improve the colloquial arabic sentiment analysis. 09 2017.

[21] Omar Alharbi. Classifying sentiment of dialectal arabic reviews: A semi-supervised approach. *International Arab Journal of Information Technology*, 16:995–1002, 04 2019.

[22] Ibtissam Touahri and Azzeddine Mazroui. Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features. *Knowledge-Based Systems*, 227:107232, 06 2021.

[23] Ashraf Elnagar, Leena Lulu, and Omar Einea. An annotated huge dataset for standard and colloquial arabic reviews for subjective sentiment analysis. *Procedia Computer Science*, 142:182–189, 2018. Arabic Computational Linguistics.

[24] Ramy Baly, Georges Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Khaled Shaban, and Wassim El-Hajj. Comparative evaluation of sentiment analysis methods across arabic dialects. *Procedia Computer Science*, 117:266–273, 11 2017.

[25] Stanford Center for Biomedical Informatics Research. Protégé.

[26] Microsoft. Visual studio code documentation. https://code.visualstudio.com/docs, 2023.

[27] Python Software Foundation. Python, 2021.

[28] TensorFlow. Tensorflow documentation.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. *scikit-learn: Machine Learning in Python*, 2021. Version 1.2.1.

[30] Charles R. Harris et al. *NumPy: Array Computing in Python*, 2021. Version 1.24.1.

[31] Wes McKinney. *pandas: powerful data analysis tools for Python*, 2020. Version 1.3.0.

[32] Taha Zerrouki. Tashaphyne, arabic light stemmer, 2012.

[33] Information Retrieval Laboratory, Center for Intelligent Information Retrieval, University of Massachusetts Amherst. ISRIStemmer: Arabic stemmer, 2013. Version 3.7.

[34] Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash.

CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France, May 2020. European Language Resources Association.

[35] Jean-Baptiste Lamy et al. *Owlready2: Ontology Library for Python*, 2021. Version 0.34.

[36] TensorFlow Documentation. Word2vec. https://www.tensorflow.org/tutorials/text/word2vec, 2023. Accessed on Date.

[37] Christopher Olah. Understanding LSTM networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015. Accessed: June 2, 2023.

[38] Long short-term memory network. https://www.sciencedirect.com/topics/computer-science/long-short-term-memory-network. Accessed: June 2, 2023.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[40] LSTM networks: A detailed explanation. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9. Accessed: June 2, 2023.

[41] GeeksforGeeks. Introduction to stemming, 2023. Accessed on 02-06-2023.

[42] Towards Data Science. The f1 score, Accessed on June 5, 2023.

[43] What is the f1 score? https://www.educative.io/answers/what-is-the-f1-score. Accessed on June 5, 2023.

[44] DeepAI. F-score. https://deepai.org/machine-learning-glossary-and-terms/f-score, Accessed on June 5, 2023.