

N° d'ordre :
N° de série :

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research



ECHAHID HAMMA LAKHDAR UNIVERSITY - EL OUED
FACULTY OF EXACT SCIENCES
Computer Science department



End of Study Thesis
Presented for the Diploma of

ACADEMIC MASTER

Domain : **Mathematics and Computer Science**
spinneret: **Computer Science**
Speciality : **Artificial Intelligence and Distributed Systems**

Presented by :

- **Cherif Rehouma**
- **Mohammed Elhabib Loubiri**

Theme

A Large Scale Hotel Arabic-Reviews Dataset

Supported in: 21- 06 - 2021 In front of jury:

Dr. Lejdel Brahime	MCA	President
M. Berdjouh Chafik	MAA	Reporter
Miss. Chourouk Guettas	MAA	Supervisor

University Year: 2020/2021

Acknowledgments

First and foremost, we thank Allah for allowing us to be in this role and for providing us with the wisdom and patience to complete this task. Second, we'd like to thank prof. Chourouk Guettas, our supervisor, for her unwavering support in terms of analysis, investigation, persistence, inspiration, and expertise. As well as her assistance, encouragement, and invaluable support during the study process and the months spent writing this thesis. In addition, we would like to thank Raki Lachrafe for his assistance and advice, as well as our parents and the friends who assisted us. We thank the rest of the discussion committee for their thoughtful feedback, inspiration, and guidance, as well as anyone who has helped us in a possible way.

Abstract

Natural language processing covers many studies and is considered the main reason for advancing techniques for understanding human behavior. Natural language processing can be used to solve problems such as plagiarism detection, extracting words and information from texts, and it is also used in machine translation and text classification. The Arabic language suffers from the lack of available large datasets for machine learning . In this work, we introduce LASHAR (A Large Scale Hotel Arabic-Reviews Dataset), the largest Hotel Reviews in Arabic Dataset for subjective sentiment analysis and machine language applications. LASHAR comprises of 1,604,762 hotel reviews collected from the Booking.com website using web scrapy, Each record contains positive or negative review text in the Arabic language, the reviewer's rating on a scale of 1 to 10 stars, and other attributes about the hotel/reviewer. We used four well-known sentiment classifiers to examine the dataset's validity and efficiency. We test the sentiment analyzers for polarity classifications. Our primary commitment is to make this benchmark data set available and open to the Arabic language research community.

Key words: Natural Language Processing, sentiment analyzers, web scrapy

ملخص

تغطي معالجة اللغة الطبيعية العديد من الدراسات وتعتبر السبب الرئيسي لتطوير التقنيات لفهم السلوك البشري. يمكن استخدام معالجة اللغة الطبيعية لحل مشكلات مثل اكتشاف السرقة الأدبية واستخراج الكلمات والمعلومات من النصوص ، كما تستخدم أيضا في الترجمة الآلية وتصنيف النص. تعاني اللغة العربية من نقص في مجموعات البيانات الكبيرة المتاحة لتطبيقات التعلم الآلي وتحليل المشاعر. وفي هذا العمل ، نقدم LASHAR (مجموعة بيانات كبيرة لمراجعات الفنادق باللغة العربية) ، وهي أكبر مراجعات الفنادق في مجموعة البيانات العربية لتحليل المشاعر الشخصية تطبيقات ولغة الآلة. يتألف LASHAR من 1,604,762 تقييمًا للفنادق تم جمعها من موقع Booking.com على الويب باستخدام آلية تجريف على شبكة الانترنت ، ويحتوي كل سجل على نص التقييم الإيجابي والسلبي باللغة العربية ، وتقييم المراجع على مقياس من 1 إلى 10 نجوم ، وسمات أخرى حول الفندق \ المراجع. استخدمنا أربعة مصنفات مشاعر معروفة لفحص صحة مجموعة البيانات وفعاليتها نقوم باختبار تحليل المشاعر لتصنيف القطبية . والتزامنا الأساسي هو جعل مجموعة البيانات المعيارية هذه متاحة ومفتوحة لمجتمع أبحاث اللغة العربية.

الكلمات المفتاحية : معالجة اللغة الطبيعية ، تحليل المشاعر ، تجريف على شبكة الإنترنت .

Résumé

Le traitement du langage naturel couvre nombreuses études et considéré comme la principale raison derrière l'avancement des techniques de compréhension du comportement humain. Le traitement du langage naturel peut être utilisé pour résoudre des problèmes tels que la détection de plagiat, l'extraction de mots et d'informations à partir de textes, et il est également utilisé dans la traduction automatique et la classification de texte. La langue Arabe souffre du manque de grands ensembles de données disponibles pour les applications d'apprentissage automatique et d'analyse des sentiments. Dans ce travail, nous présentons LASHAR (A Large Scale Hotel Arabic-Reviews Dataset), le plus grand ensemble de données d'évaluation d'hôtels en Arabe pour l'analyse des sentiments subjectifs et ses applications. LASHAR comprend 1 604 762 avis sur les hôtels collectés à partir du site Web Booking.com. Chaque enregistrement contient le texte de l'avis en Arabe, la note de l'évaluateur sur une échelle de 1 à 10 étoiles et d'autres attributs concernant l'hôtel / l'évaluateur. Nous avons utilisé quatre classificateurs de sentiments bien connus pour examiner la validité et l'efficacité de l'ensemble de données. Nous testons les analyseurs de sentiment pour les classifications de polarité. Notre principal engagement est de rendre cet ensemble de données de référence disponible et ouvert à la communauté de recherche en langue Arabe.

Mots clés : Traitement du langage naturel, analyseurs de sentiments, web scrapy

CONTENTS

List of Figures	iv
List of Tables	vi
List of Abbreviations	vii
General Introduction	1
1 Natural Language Processing	3
1.1 Introduction	4
1.2 Natural Language Processing	4
1.2.1 Definition of Natural Language Processing (NLP)	4
1.2.2 History of Natural Language Processing (NLP)	4
1.2.3 Techniques and methods of natural language processing	4
1.2.4 Levels of natural language analysis	5
1.2.5 Applications of Natural Language Processing (NLP)	5
1.3 Arabic natural language processing (ANLP)	6
1.3.1 Arabic language	6
1.3.2 Arabic language characteristics	6
1.3.3 Arabic language complexity	7
1.3.4 Arabic sentiment analysis	8
1.4 Literature Review	8

1.4.1	Definition of dataset	8
1.4.2	Definition of corpus	8
1.4.3	Related works	8
1.4.4	Comparison between the different datasets	13
1.5	Evaluation	14
1.6	Conclusion	14
2	web scraping	15
2.1	Introduction	16
2.2	Definition of web scraping	16
2.2.1	Web scraping architecture	17
2.3	Tools of web scraping	17
2.3.1	XPath and CSS selectors	17
2.3.2	Scrapy	17
2.3.3	Web crawler	18
2.3.4	Splash - A javascript rendering service	18
2.3.5	Selenium Web Driver	18
2.4	Choosing tools	19
2.5	The Steps of creating our scraper	19
2.5.1	Install selenium and download the driver of the browser	19
2.5.2	Diagram of scraper	20
2.5.3	Login to the website	20
2.5.4	Choosing country and access to the hotels	22
2.6	Data preview	31
2.6.1	Image of the data collection	31
2.6.2	Time spent	31
2.6.3	Statistics on LASHAR dataset	32
2.7	Conclusion	32
3	Sentiment Analysis	33
3.1	Introduction	34
3.2	Data reformatting	34
3.3	Text pre-processing	35

3.3.1	Library import and Arabic diacritics	35
3.3.2	Function preprocess text	36
3.3.3	Example of pre-processing text	36
3.3.4	Natural Language Toolkit	37
3.4	Google collaboratory (colab)	37
3.5	Arabic sentiment analysis algorithms	37
3.5.1	Logistic Regression	37
3.5.2	Random Forest Classifier	38
3.5.3	Naive Bayes Classifier (Multinomial)	39
3.5.4	Support Vector Machine	40
3.5.5	Evaluation of the used algorithms	41
3.6	Comparison between LASHAR and the different datasets	42
3.7	Conclusion	42
	general Conclusion	43
	References	44

LIST OF FIGURES

1.1	ASTD tweets examples.[18]	9
1.2	Sample reviews in MSA and DA [20]	11
2.1	Web scraping architecture	17
2.2	Selenium web driver architecture	19
2.3	Diagram of scraper	20
2.4	Access to the website	21
2.5	The countries	21
2.6	Scroll down	22
2.7	All countries of Booking.com	22
2.8	Choosing a country	23
2.9	Best cities of Egypt	23
2.10	Choosing a city	24
2.11	Hotels of Sharm El-Sheikh	24
2.12	First part of access_to_hotel()	25
2.13	Second part of access_to_hotel()	26
2.14	Third part of access_to_hotel()	26
2.15	Reviews	27
2.16	Arabic reviews	27
2.17	First part of class reviews()	29
2.18	Second part of class reviews()	29
2.19	First part of function get_reviews()	30

2.20	Second part of function <code>get_reviews()</code>	30
2.21	Some reviews of Abdeen Palace Hostel	31
3.1	Example of the latest data	34
3.2	Formatting the dataset	35
3.3	Library import and arabic diacritics	35
3.4	Preprocess function	36
3.5	Text before	36
3.6	Text after	36
3.7	Function of logistic regression	38
3.8	Result of logistic regression	38
3.9	Function of random forest classifier	39
3.10	Result of random forest classifier	39
3.11	Function of naive Bayes classifier	40
3.12	Result of naive Bayes classifier	40
3.13	Function of support vector machine	40
3.14	Result of support vector machine	41

LIST OF TABLES

1.1	Important Dataset Statistics.[17]	9
1.2	Twitter dataset statistics	9
1.3	Statistics on HARD dataset.[20]	10
1.4	Individual and Aggregated Classes Distribution Over the Dataset.	12
1.5	Comparison between the articles	13
2.1	Statistics on LASHAR dataset	32
3.1	comparison of Colab and Colab pro[33]	37
3.2	Comparison between the algorithms	41

enumitem

abbrvitemize1 [abbrv,1]label=,labelwidth=1in,align=parleft,itemsep=0.1leftmargin=!

LIST OF ABBREVIATIONS

NLP Natural Language Processing

ANLP Arabic natural language processing

SA sentiment analysis

ASA Arabic sentiment analysis

NLTK Natural Language Toolkit

LABR A Large Scale Arabic Book Reviews Dataset

ASTD Arabic Sentiment Tweets Dataset

ALC Arabic Learner Corpus

HARD Hotel Arabic Reviews Dataset

ATSAD Arabic Tweets Sentiment Analysis Dataset

AVTCD Towards a corpus of violence acts in Arabic social media

GENERAL INTRODUCTION

In machine learning era, valuable data is the key to reach important deductions. In fact, colossal amounts of data are required with the fast development of machine algorithms and hardware. One of the most critical fields in machines cognition is Natural Language Processing (NLP). NLP has a long history. But we are mainly interested in the part that started when the field began exchanging expertise with other fields such as computer science and artificial intelligence. With the help of programming languages and neural networks, NLP aims to give machines the ability to understand and use words and sentences like a human being does. Since, the machine has a more accurate computational performance, it can evaluate the basic units of the language, implement the abstraction in the best ways possible and provide other language utilities to the world, including machine translation, information retrieval and emotional detection (sentiment analysis).

English is the language of the world no doubt. Thus, it had the biggest share of interest in NLP field researches and contributions. It also has many advantages that encourage academic researchers, especially data abundance. Things does not go that trouble-free when speaking about the spoken tongue of one billion and seven hundred million of the world population, Arabic. Due to the lack of contributors and valuable resources of data that touch profitable areas, This language did not get the same amount of interest.

In this research project we give high priority to the task of data collection and the goal of enhancing Arabic NLP with a contribution of a polished corpus that is mainly directed to sentiment analysis use and can also be used in developing generic pre-trained

models. In order to do so, we need to be comfortable with digging a fast growing renewable resource such as web 2.0. This so called web scraping functionality is about implementing problem solving techniques for automating the navigation between web servers, recognizing and collecting the targeted data.

The rest of this thesis is organized as follows. In Chapter 1, we present the State of the Art for NLP, Arabic NLP, Sentiment Analysis and the latest contributions concerning Arabic data collection. In Chapter 2, we construct and examine a web scraping application for one of the biggest assets about reviews “booking.com”. In Chapter 3, we present a preprocessed corpus of over a million Arabic reviews and its evaluation and testing with some famous machine learning algorithms. Finally, the conclusion and future vision especially about crawling other Arabic big data resources.

CHAPTER 1

NATURAL LANGUAGE PROCESSING

1.1 Introduction

Natural language processing, as a branch of AI, is about giving computer the ability to understand text and spoken words in much the same way human beings can. In NLP the basic units are words and the implementation of any system that processes a language depends and relies on words or tokens in its work. In this chapter, we are interested in introducing and presenting briefly natural Language Processing and talk in more details about Arabic natural Language Processing.

1.2 Natural Language Processing

1.2.1 Definition of Natural Language Processing (NLP)

Natural language processing's key purpose is to create computers that can understand text or voice input, and then respond with text or speech of their own, much like humans. The way of NLP works is by combining statistical, machine learning, and deep learning models and computational linguistics rule-based modeling of human language. These technologies, when used together, enable computers to process text or speech data in the context of human language and 'understand' its full context, including the speaker or writer's intent and emotion.[1]

1.2.2 History of Natural Language Processing (NLP)

NLP can be traced all the way back to the 1950s. Alan Turing published an essay in 1950 titled "Computing Machinery and Intelligence" in which he introduced the Turing test as a measure of intelligence, a challenge that requires the automated interpretation and generation of natural language, but was not expressed as an issue separate from artificial intelligence.[2]

1.2.3 Techniques and methods of natural language processing

✓ **Sentence Segmentation :** The method of separating the printed text into comprehensible chunks, such as phrases, sentences, or subjects, is known as sentence segmentation.

✓ **Tokenization:** Tokens are the names given to each of these smaller units. That units are the result of breaking down a paragraph, sentence, or whole text document.

✓ **Stemming:** Stemming is a technique for deleting a word's suffix and reducing it to its root word.

1.2.4 Levels of natural language analysis

Natural Language Processing works on multiple levels and most often, these different areas synergize well with each other

- 1- Phonetic or phonological level: deals with pronunciation
- 2- Morphological level: works with the meaning-carrying bits of sentences, as well as suffixes and prefixes.
- 3- Lexical level: works with a word's lexical context.
- 4- The syntactic level is concerned with sentence form and grammar.
- 5- Semantic level: this level is concerned with the interpretation of terms and sentences.
- 6- Discourse level: this level is concerned with the arrangement of various types of texts.
- 7- Pragmatic level: this level deals with information that emerges from the outer world, i.e. knowledge that is not included within the document's text.

1.2.5 Applications of Natural Language Processing (NLP)

The techniques of automatic language processing can be applied to a multitude of fields such as machine translation, information extraction, automatic summary of texts, question and answer systems etc. The most frequent applications of NLP are:

✓ **Machine Translation:** Machine Translation is the procedure of automatically converting the text in one language to another language while keeping the meaning intact.

✓ **Information extraction:** Information extraction is the process of extracting information from unstructured textual sources to enable finding entities as well as classifying and storing them in a database.

✓ **Question answering (QA):** Works for a building system that can communicate with humans for automatically giving correct answers to the questions.

✓ **Automatic Text Summarization :** Summarization is a process that must

make the size of the text, the smallest one while keeping the same meaning of the text and save the preserving key informational elements.

1.3 Arabic natural language processing (ANLP)

Arabic natural language processing (ANLP) has attracted many researchers after significant research has been carried out on English NLP and that of other languages. Many ANLP laboratories have been established. Recently, ANLP has received more attention, and several applications have been developed including text categorization, web page spam detection, and sentiment analysis [1]–[2]. However, owing to two major challenges, creating ANLP tools would take more time and effort: combining letters in the Arabic language and removing diacritics that indicate vowels.[9].

1.3.1 Arabic language

Arabic is an Afro-Asiatic language that developed in the Middle East. More than 250 million individuals speak the Arabic language across the world [10]. Islam is helped tremendously in the spread rapidly of the Arabic language, Knowing that the Arabic language is had exist for a long time before the coming of Islam, estimated to be 1.5 billion people [10]. Historically speaking, Arabic is rooted in Classical Arabic (CA), and has been used by Arab communities' native language since 600 AD. It is associated with Islam and the Quran. However, over the centuries, the language has evolved and been simplified to create what is known as Modern Standard Arabic (MSA). The terminology and the linguistic features of MSA differ from those of CA, The form of words and sentences, though, has not changed. In addition to CA and MSA, each region has a dialect of Arabic spoken in the community (between friends and family) [11].

1.3.2 Arabic language characteristics

The Arabic language consists of grammar, spelling, punctuation marks, slang as informal language, idioms, and pronunciation. Many characteristics make the Arabic language distinctive [5]:

1. reading and writing in Arabic moves from right to left.
2. The language consists of 28 characters.

3. In Arabic, upper and lower cases are not distinguished, like Chinese, Japanese, and Korean.
4. Numbers are divided into plural, dual, and singular, with two genders—feminine and masculine.
5. The language comprises several words formed from roots, and several root words are composed of three letters.
6. Verbs in the past tense are identified by suffixes, and verbs in the present or future tenses are designated by prefixes; for example, “dahabat” means “she went,” but “tadhabu” means “she goes.”
7. Sentences start with verbs, followed by subjects, and are finished with objects for the predicate
8. Arabic tolerates the deletion of subject pronouns (pro-drop language) like Italian and Chinese [11]

1.3.3 Arabic language complexity

There are some factors that stymie development in Arabic Natural Language Processing (NLP) as opposed to English and other European languages. There are some of them:

- Since Arabic is heavily inflectional and derivational, morphological research is a difficult challenge.
- Since most words are not represented by diacritics in written language, uncertainty arises, necessitating the use of complex morphological rules to define tokens and decipher the text.
- The writing direction is from right-to-left and some of the characters change their shapes based on their location in the word.
- Capitalization is not used in Arabic, which makes it hard to identify proper names, acronyms, and abbreviations.

Aside from the linguistic problems mentioned above, there is a dearth of Arabic corpora, lexicons, and machine-readable dictionaries, both of which are essential for study in various fields.[12]

1.3.4 Arabic sentiment analysis

Sentiment Analysis is the process of identifying and extracting subjective knowledge from a piece of writing using Natural Language Processing and Machine Learning. Also, People's thoughts, feelings, assessments, behaviors, and feelings towards institutions such as goods, services, groups, people, concerns, activities, topics, and their characteristics are analyzed by analyzing feelings. This allows an understanding of the general public's opinions or attitudes on specific topics, products, or services.[13]

1.4 Literature Review

1.4.1 Definition of dataset

A collection of data that is treated as a single unit by a computer.[14] The dataset contains a lot of separate pieces of data but can be used to train an algorithm with the goal of finding predictable patterns inside the whole dataset.[15]

1.4.2 Definition of corpus

A corpus is a collection of authentic text or audio organized into datasets. 'Authentic' in this case means text written or audio spoken by a native of the language or dialect. A corpus can be made up of everything from newspapers, novels, recipes, and radio broadcasts to television shows, movies, and tweets.

In natural language processing, a corpus contains text and speech data that can be used to train AI and machine learning systems.[16]

1.4.3 Related works

1 LABR: A Large Scale Arabic Book Reviews Dataset

This dataset created by Mohamed Aly ,et al [17], contains over 63k book reviews in the Arabic language. Figure 01 shows some important facts about the dataset. They have applied two tasks to the dataset :

- Sentiment polarity classification: The aim is to guess whether the evaluation will be positive or negative.
- Rating classification: where the aim is to forecast the review's rating.[17]

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

Table 1.1: Important Dataset Statistics.[17]

2 ASTD: Arabic Sentiment Tweets Dataset

Created by Mahmoud Nabil, et al in 2015 [18], the ASTD: Arabic Sentiment Tweets Dataset Dataset contains over 10k Arabic sentiment tweets classified into 4 classes: subjective positive, subjective negative, subjective mixed, and objective., in the Arabic language. They used the dataset for sentiment polarity classification using a wide variety of standard classifiers to do four-way sentiment classification after doing standard partitioning.

Total Number of conflict free tweets	10,006
Subjective positive tweets	799
Subjective negative tweets	1,684
Subjective mixed tweets	832
Objective tweets	6,691

Table 1.2: Twitter dataset statistics

	Tweet	Translation	Rate
1	أكثر شعور بوجع ! ^ #لما تجوع في بيت مو بيتكم ☹️	Feeling that hurts ^ ! #To starve in a house not yours	Negative
2	محبين البرنامج يزيدوا :)	Fans of El-Bernameg are increasing :)	Positive
3	#كفاية اسفاه	#stop smallness	Negative
4	الطاقة البشرية اذا ما احسن استغلالها هي رصيدها وليست عبئا قوتنا في عددا	Human energy if properly exploited is an asset and not a burden our strength in our numbers	Positive
5	احسن الشيخ حسن عبد البصير امام مسجد سيدى جابر الذي رفض تعليمات الأوقاف بنفاق مريسي في خطبة الجمعة تعلموا الاستقامة أيها #الاخوان الكاذبون	I greet Sheikh Hassan AbdelBassir Imam Sidi Gaber mosque, who refused the instructions of the endowments to hypocrite Morsi in his Friday sermon learn the integrity liars brotherhood	Mixed
6	هل تنجح ايتكو مدريد بلقب الليجا الأحد القادم؟ #برشلونة	Is Atletico Madrid going to be crowned La Liga next Sunday? # Barcelona	Objective

Figure 1.1: ASTD tweets examples.[18]

3 Arabic Learner Corpus (ALC)

A collection of written and spoken materials produced by learners of Arabic in Saudi Arabia. The ALC contains 0.2 million Arabic words.[19]

4 Hotel Arabic Reviews Dataset (HARD)

The dataset is created by Ashraf Elnagar, et al. [20], and is a collection of 373,772 Arabic reviews. They have applied some algorithms of sentiment analysis on the dataset: Logistic regression, Passive-aggressive, SVM, Perceptron, Random forest, AdaBoost.

Title	Number	Title	Number
Number of reviews	373,772	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

Table 1.3: Statistics on HARD dataset.[20]

Rating	Arabic Type	Book ID	Review	#
5	MSA	1468	فندق رائع. الفندق مميز بموقعه وتصميمه وديكورات غرفة السرير واسع ومريح جدا. لا شيء	1
1	MSA	1468	استغرب تقييم الفندق 5 نجوم . لا شيء يستحق 2 نجمة	2
2	MSA	1514	ضعيف كل شيء خرابان ووسخ	3
1	DA	2125	جيد موقعه قريب من الاسواق السرير مموّج سعره موب مناسب عالغرفة	4
4	DA	1167	جيد كل شيء النت بفلوس الفرشة موب نظيفة	5
3	DA	1542	مقبول اعجيني ان السعر السعر مناسب ولكن مافية خصوصية تسمح الي ساكن جنبك او في الطابق اللي فوقك بترطع ماكو نوم عدل من الإزعاج	6

Figure 1.2: Sample reviews in MSA and DA [20]

5 Arabic Tweets Sentiment Analysis Dataset (ATSAD)

The dataset is created by Kathrein Abu Kwaik, et al. [21] They used two well-known methodologies to test the Tweets corpus: intrinsic and extrinsic evaluations : In intrinsic evaluation, the corpus is directly evaluated in terms of its accuracy and quality. In extrinsic evaluation, the dataset is going to be assessed with respect to its impact on an external task which in this case is the sentiment analysis model.

6 Towards a corpus of violence acts in Arabic social media

Created by Ayman Alhelbawy, et al. in 2016. [22] Annotated corpus of Arabic tweets which mention a violence act. It is based on a collection on 20,000 tweets which is manually annotated using the crowdflower platform. They made the dataset available for classification on CrowdFlower, a common crowdsourcing website. At least five different contributors classified each tweet text.

Violence Class	Individuals		Aggregate	
	Count	%	Count	%
crises	4,066	3	274	1
violence	6,823	4	487	2
accident	5,679	4	558	3
crime	10,942	7	1,331	7
HRA	19,079	12	2,367	12
conflict	23,555	15	3,189	16
opinion	29,556	19	4,261	21
other	56,834	36	7,684	38

Table 1.4: Individual and Aggregated Classes Distribution Over the Dataset.

1.4.4 Comparison between the different datasets

	LABR	ASTD	ALC	HARD	ATSAD	AVTCD
Size of the dataset	63,000 Arabic reviews	10,000 Arabic tweets	240,000 words	490,587 reviews	36,868 tweets	557,576 tweets
Year of collecting the data	2013	2015	2014	2018	2019	2016
Content	Text	Text	Text	Text	Text	Text
Source	goodreads.com	socialbakers.com/twitter.com twitter.com/EgyptTrends	Educational institutions	Bookin-g.com	Twitter	Twitter
Domain	book reviews	Tweets	education of Arabic	Hotels reviews	tweets	tweets
Scraping method	Did not mention	Amazon Mechanical Turk (AMT)	manually annotated	Did not mention	python +twitter API	manually and using crowdflow-er platform

Table 1.5: Comparison between the articles .

1.5 Evaluation

Like table 1.5 shows in the row “size of the dataset” that the datasets in all referenced works are not large enough. In the aim of creating a larger scale dataset for Arabic language and test its quality by using it in sentimental analysis, this work was presented.

1.6 Conclusion

In this chapter, we presented an overview of the most important Arabic semantic dataset Used in natural language processing and literature review of collection data. In the next chapter, we will present our script of how we get the reviews.

CHAPTER 2

WEB SCRAPING

2.1 Introduction

One of the most important assets that can be found on the internet is information. To function properly, many web servers need a significant volume of data. Online search engines (Google, Duckduckgo), commodity pricing and specification compare, and internal competition and competitor research methods are examples of those applications. Those programs typically use Web Scraping to retrieve information from the internet and convert it into useful formatted data.

2.2 Definition of web scraping

Web scraping is a process that automatically collects and extracts data from the internet. Web scraper is one of the most efficient tools to extract data from websites. The purpose of web scraping is to extract a large quantity of data and save it to a local environment . In the early days, the only way to extract data from websites was by copy-pasting what one saw on the website. Web scraping is becoming a popular technique as it allows new startups to quickly obtain large amounts of data. One of the most typical examples of web scraping are price comparison and reviews of websites.[23]

2.2.1 Web scraping architecture

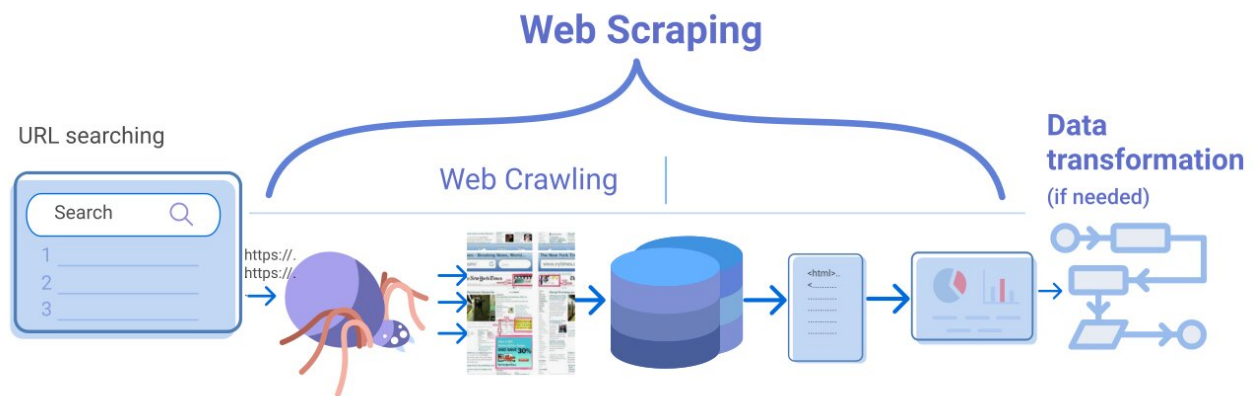


Figure 2.1: Web scraping architecture

2.3 Tools of web scraping

2.3.1 XPath and CSS selectors

Xpath is a powerful query language for fetching nodes from an XML document. Because HTML is a subset of XML, XPath can also be used to fetch an HTML element.[25] Even though XPath is powerful, it is complicated and difficult to implement. The CSS selector is a more human-friendly alternative which allows selecting the HTML element based on the CSS styles, class name, or id associated with the element. [26]. In CSS, selectors are used to target the HTML elements on our web pages that we want to style. There are a wide variety of CSS selectors available, allowing for fine-grained precision when selecting elements to style. [26]

2.3.2 Scrapy

Scrapy is a fast high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages. It can be used for a wide range of purposes, from data mining to monitoring and automated testing.[27]

2.3.3 Web crawler

A Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web, typically operated by search engines for the purpose of Web indexing (web spidering).[28]

2.3.4 Splash - A javascript rendering service

Splash is a javascript rendering service. It's a lightweight web browser with an HTTP API, implemented in Python 3 using Twisted and QT5. The (twisted) QT reactor is used to make the service fully asynchronous allowing to take advantage of webkit concurrency via QT main loop [29]

2.3.5 Selenium Web Driver

Selenium is a library that exposes interface to control real Web Browser automatically . There are some websites that heavily rely on JavaScript and can only be scraped with real browser. Even though Selenium makes it possible to scrape some complicated websites, it consumes a lot more computer resources. Selenium also slows down the Scraping process considerably because it needs to open the browser and loads the entire web page. [30]

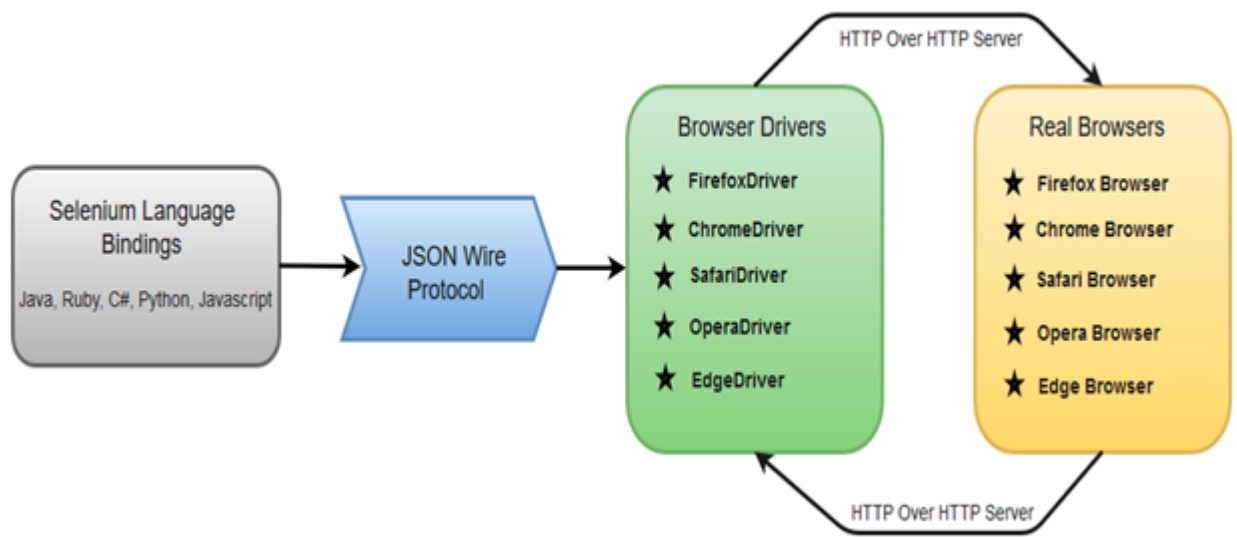


Figure 2.2: Selenium web driver architecture

2.4 Choosing tools

After we chose the website (Booking.com) that we will be working on it, we found out that this website relied heavily on JavaScript; so there are two ways: either to use splash or selenium. After we compared both of them, we decided to choose selenium, because is really easier to use and faster.

2.5 The Steps of creating our scraper

2.5.1 Install selenium and download the driver of the browser

Downloading python bindings for selenium We use pip to install the selenium package:
 pip install selenium

After we install selenium we need drivers because selenium requires a driver to interface with the chosen browser. Chrome, for example, requires chromedriver we find in this link : <https://sites.google.com/a/chromium.org/chromedriver/downloads>

2.5.2 Diagram of scraper

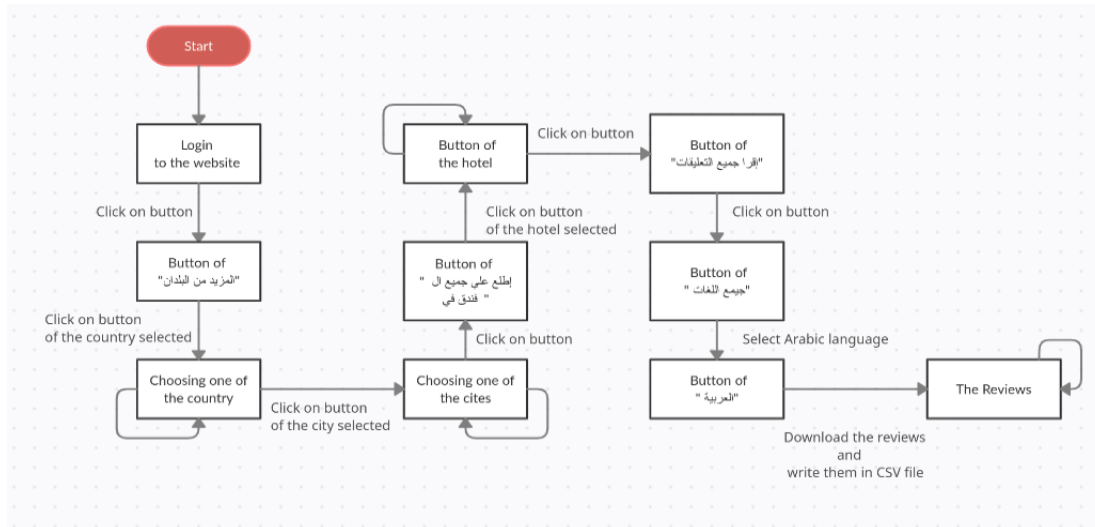


Figure 2.3: Diagram of scraper

2.5.3 Login to the website

Define the driver path of chrome and access to the website in selenium In figure 2.4: line 12 we define the driver and in line 15 we access to the website (<https://www.booking.com/index.ar.html>) We use try and except WebDriverException for if the internet connection is lost the scraper won't stop, it refreshes many time until the internet is back. In line 26 we edit the size of windows because the default windows doesn't work with the full screen mode.


```

1  from selenium import webdriver
2  from selenium.common.exceptions import NoSuchElementException
3  from selenium.common.exceptions import WebDriverException
4  from selenium.common.exceptions import StaleElementReferenceException
5  from selenium.common.exceptions import NoSuchWindowException
6  from selenium.common.exceptions import ElementClickInterceptedException
7  from selenium.common.exceptions import ElementNotInteractableException
8  import time
9  from get_reviews import reviews as the_reviews
10
11 # define the driver path of chrome
12 driver = webdriver.Chrome(executable_path="./chromedriver")
13 # launch the website
14 try :
15     driver.get("https://www.booking.com/index.ar.html")
16 except WebDriverException :
17     login = False
18     while login == False :
19         driver.refresh()
20         try:
21             driver.get("https://www.booking.com/index.ar.html")
22             login = True
23         except WebDriverException :
24             login = False
25 # edit the size of windows
26 driver.set_window_size(1366,768)

```

Figure 2.4: Access to the website

After accessing the website, we need to scroll down to display all the countries as shown in figure 2.5

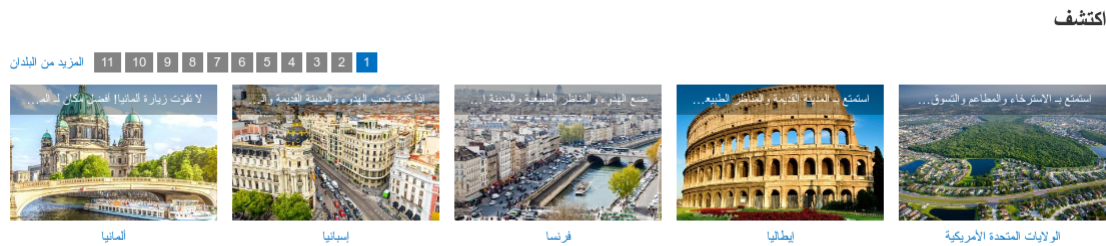


Figure 2.5: The countries

So we script this in line 102 of figure 2.6 now we click in (المزيد من البلدان) we find it in figure 2.5 , scripted from line 104 to line 110 in figure 2.6 with find_element_by_xpath() function gives the XPath of this element (المزيد من البلدان) from HTML markup of the website and click to this element with click() function

```

101 # scroll down in main page og booking.com
102 driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")
103 # to see more countries
104 try:
105     countries = driver.find_element_by_xpath("//a[@class='dcbi-more']")
106     countries.click()
107 except NoSuchElementException :
108     time.sleep(2)
109     countries = driver.find_element_by_xpath("//a[@class='dcbi-more']")
110     countries.click()

```

Figure 2.6: Scroll down

Now we can see all the countries of this website like shown in figure 2.7

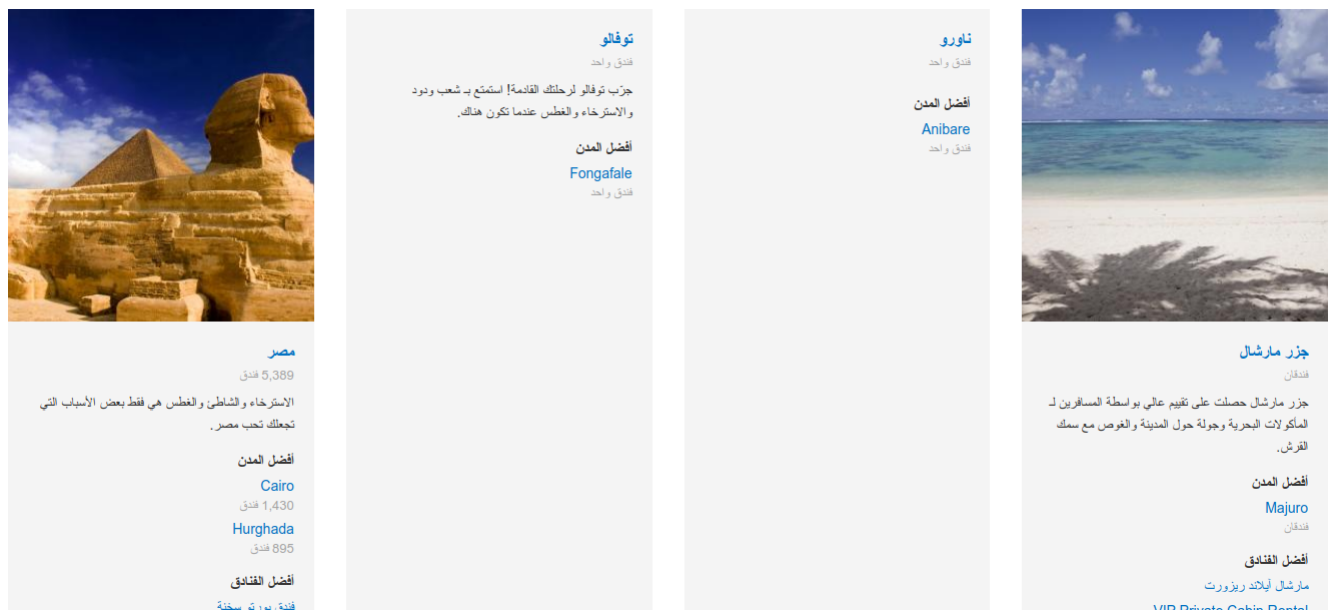


Figure 2.7: All countries of Booking.com

2.5.4 Choosing country and access to the hotels

In figure 2.8 in line 113, we give the driver XPath of the element and in line 114 the driver clicks on it. The result appears in figure 2.9

```

111 # chosing a country
112 try:
113     egybt = driver.find_element_by_xpath("(//div[@class='bui-u-full-height dci-country__container'])[64]/div//a")
114     egybt.click()
115 except NoSuchElementException:
116     time.sleep(2)
117     egybt = driver.find_element_by_xpath("(//div[@class='bui-u-full-height dci-country__container'])[64]/div//a")
118     egybt.click()

```

Figure 2.8: Choosing a country

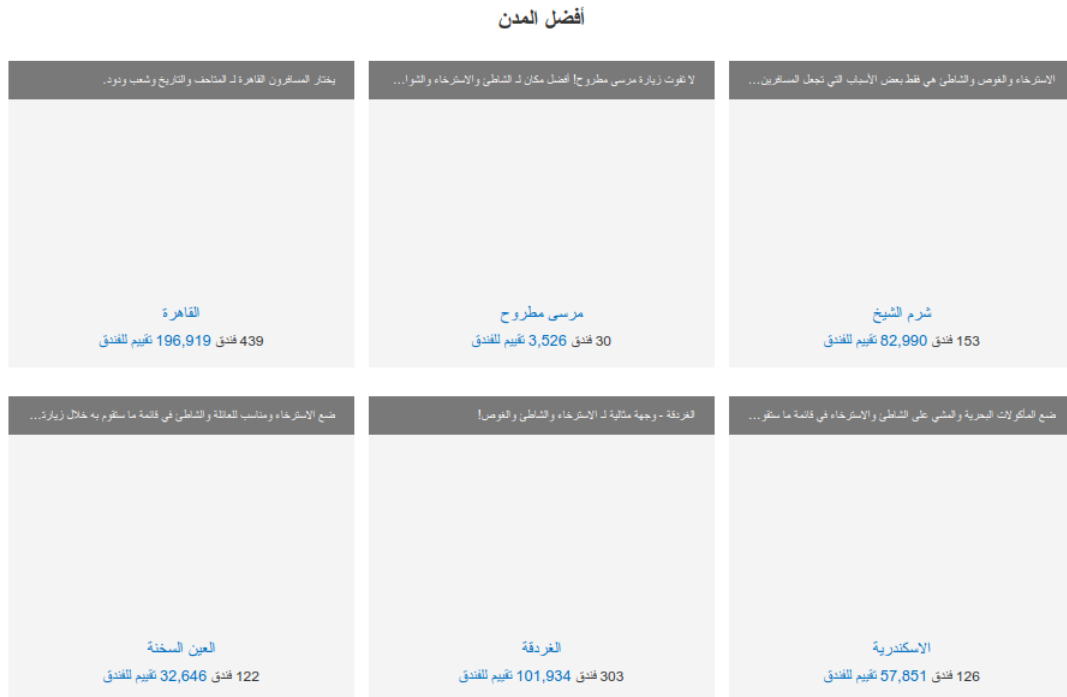


Figure 2.9: Best cities of Egypt

Choose one city and click on it in figure 2.10, we create a “for” for looping and access to the six cities. In line 123 we give the driver the elements of the city according to the value of “b” and in line 124 we click on it

```

119 # chose best city in egypt
120 best_citys_link = driver.current_url
121 for b in range(1,7):
122     # see the hotel of city
123     hotel_city = driver.find_element_by_xpath(f"(.//h3[@class='dci-country-popular-item__title']/a)[{b}]")
124     hotel_city.click()
125     # see all the hotel of city
126     all_hotel = driver.find_element_by_xpath("//div[@class='lp-bui-section bui-spacer--largest x2']/a")
127     all_hotel.click()
128     # call acssece hotel function for get the reviews
129     acssece_hotel()
130     # get the next list of hotel
131     window_after = driver.window_handles[0]
132     driver.switch_to.window(window_after)
133     driver.set_window_size(1366, 768)
134     try:
135         paging_next = driver.find_element_by_xpath(
136             "//ul[@class='bui-pagination__list']/li[@class='bui-pagination__item bui-pagination__next-arrow']/a")
137     except NoSuchElementException:
138         print('nothing')
139     while paging_next:
140         paging_next.click()
141         acssece_hotel()
142         window_after = driver.window_handles[0]
143         driver.switch_to.window(window_after)
144         driver.set_window_size(1366, 768)
145         paging_next = driver.find_element_by_xpath(
146             ("//ul[@class='bui-pagination__list']/li[@class='bui-pagination__item bui-pagination__next-arrow']/a")
147         driver.get(best_citys_link)

```

Figure 2.10: Choosing a city

After that, we click on the button that has a label

(اطلع على جميع الـ ١٨٩ فندق في شرم الشيخ) to see all the hotels of Sharm El-Sheikh. This action is done with the commands in lines 126 and 127 in figure 2.10. Now we need to call the function `access_to_hotel()`, all code of this function is shown through three figures (Figure 2.12, Figure 2.13 ,Figure 2.14)

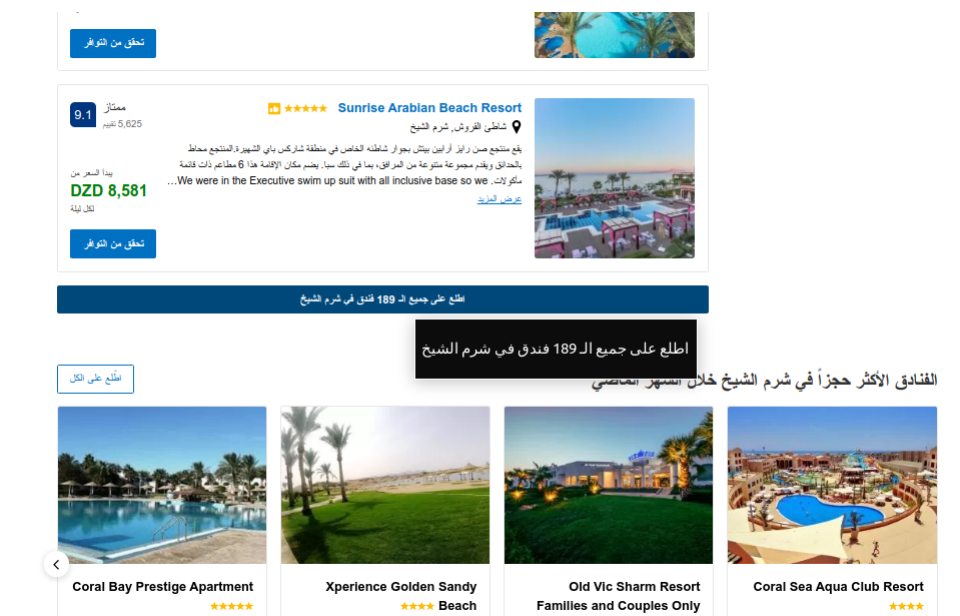


Figure 2.11: Hotels of Sharm El-Sheikh

In figure 2.12, line 32 we put all the hotels that the web page displays in it the value called “kind_hotels_link” and create a “for” for looping with the same command for each hotel In line 37 we chose one hotel and click on it and then we use try and except in case the code doesn’t execute.

We need to get the name of the hotel that we click on it with the command in figure 2.13 Form the line 58 to 66, now we are going to see all reviews by clicking in the button that have the label (” اقرأ جميع التقييمات ”) in figure 2.14, this action can be done through code lines from 68 to 75 in figure 2.13. Finally, we should choose the Arabic language in order to see just the Arabic reviews, doing that in code lines from 77 to 91, see figure 2.14

```

27 # function for access to all the hotel of city
28 def access_to_hotel():
29     # save link of this page in value
30     all_hotel_link = driver.current_url
31     # now we chose hotel to get in
32     kind_hotel_link = driver.find_elements_by_xpath("//div[@class='sr_property_block_main_row']/div/div/h3/a")
33     for khl in range(0, len(kind_hotel_link)):
34         staleElement = True
35         while staleElement:
36             try:
37                 kind_hotel_link[khl].click()
38                 staleElement = False
39             except NoSuchElementException:
40                 window_after = driver.window_handles[0]
41                 driver.switch_to.window(window_after)
42                 driver.set_window_size(1366, 768)
43                 staleElement = True
44                 kind_hotel_link = driver.find_elements_by_xpath(
45                     "//a[contains(@class,' sr_item_photo_link sr_hotel_preview_track')]")
46             except ElementClickInterceptedException:
47                 staleElement = True
48                 kind_hotel_link = driver.find_elements_by_xpath(
49                     "//a[contains(@class,' sr_item_photo_link sr_hotel_preview_track')]")
50             except StaleElementReferenceException:
51                 staleElement = True
52                 kind_hotel_link = driver.find_elements_by_xpath(
53                     "//a[contains(@class,' sr_item_photo_link sr_hotel_preview_track')]")
54             except ElementNotInteractableException:
55                 driver.refresh()
56                 staleElement = True

```

Figure 2.12: First part of access_to_hotel()

```

57 # get name of the hotel
58 try:
59     name_of_hotel = driver.find_element_by_xpath("//div[@id='wrap-hotelpage-top']/a[contains(@class,'fn ')]").get_attribute(
60         ("innerHTML")
61     except NoSuchElementException:
62         window_after = driver.window_handles[1]
63         driver.switch_to.window(window_after)
64         name_of_hotel = driver.find_element_by_xpath("//div[@id='wrap-hotelpage-top']/a[contains(@class,'fn ')]").get_attribute(
65             ("innerHTML")
66 name_of_hotel_file = f'{name_of_hotel}.csv'
67 # see the all reviews
68 try:
69     all_reviews = driver.find_element_by_xpath(
70         ("//div[@class='hp-featured_reviews-bottom']/button[contains(@class, 'bui-button bui-button--secondary')]")
71     all_reviews.click()
72 except ElementNotInteractableException:
73     driver.refresh()
74 except:
75     print("can't find the element of show all reviews ")

```

Figure 2.13: Second part of access_to_hotel()

```

76 # chose language of reviews
77 try:
78     reviews_language = driver.find_element_by_xpath("//div[@id='review_lang_filter']/button")
79     reviews_language.click()
80 except ElementNotInteractableException:
81     driver.refresh()
82 except:
83     print("can't find the element for chosing language of reviews")
84 # chose arabic reviews
85 try:
86     arab_reviews = driver.find_element_by_xpath("(//ul[@class='bui-dropdown-menu__items'])[8]/li[2]/button")
87     arab_reviews.click()
88 except ElementNotInteractableException:
89     driver.refresh()
90 except NoSuchElementException:
91     print("can't find the element of arabic reviews ")
92 driver.set_window_size(1366, 768)
93 time.sleep(3)
94 try:
95     link_reviews = driver.find_element_by_xpath("//a[@class='pagenext']").get_attribute('href')
96 except NoSuchElementException:
97     print("can't find the element for get the link of reviews ")
98 # now call get_reviews class for get reviews
99 try:
100     the_reviews().Reviews(link_reviews, name_of_hotel_file)
101 except:
102     pass
103 driver.close()

```

Figure 2.14: Third part of access_to_hotel()

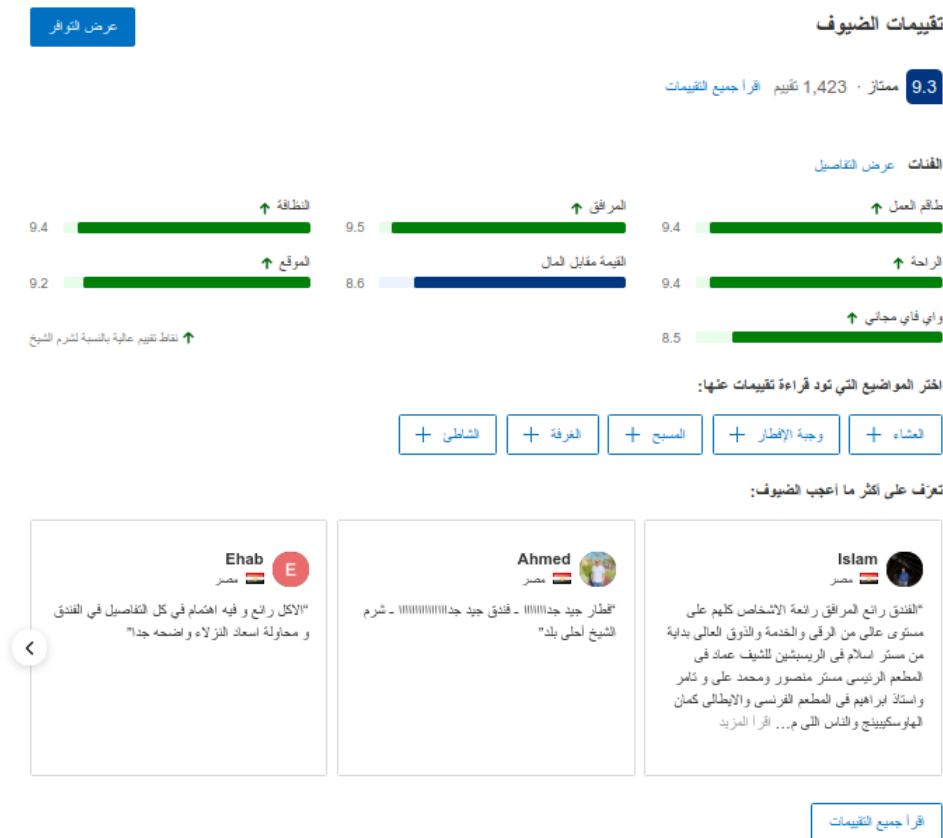


Figure 2.15: Reviews

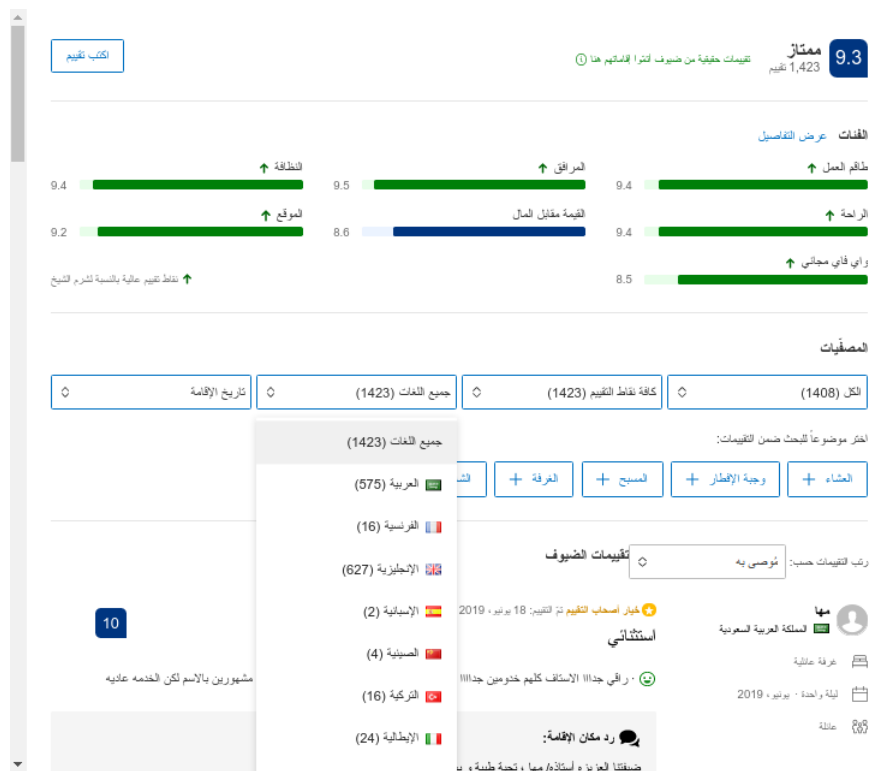


Figure 2.16: Arabic reviews

After we see the reviews and get the link to them and the name of the hotel, we retrieve them from the website. To do that, we create a class reviews () and call it in line 100 in figure 2.14 . In figure 2.17, line 15, we login to the link of reviews to collect this reviews we need a CSV file to save them in it, we create a file as shown in figure 2.17 in lines 26 to 29 To fill this file we call the function get_reviews() figure 2.19 Each person has a “username”, “nationality”, “personal score”, “review title”, “positive part”, “negative part”, “date of reviews”. All these information are found in a div. Next, the div is given to the driver to work on it with xpath as in figure 2.19 line 33. We made a “for” for looping the same code for each div and get all the information of this person like we get the username in figure 2.19 line 37. After determining the element of username, get_attribute(“innerHTML”) function is applied to get the value of this element.


```

1 from selenium import webdriver
2 from selenium.webdriver.chrome.options import Options
3 from selenium.common.exceptions import NoSuchElementException
4 from selenium.common.exceptions import WebDriverException
5 import time
6 import csv
7 import os
8 class reviews() :
9
10     def Reviews(self,link_reviews,name_of_Hotal_file):
11         chrome_options = Options()
12         chrome_options.add_argument("--headless")
13         driver = webdriver.Chrome(executable_path='./chromedriver')
14         try :
15             driver.get(link_reviews)
16         except WebDriverException :
17             login = False
18             while login == False :
19                 driver.refresh()
20                 try:
21                     driver.get(link_reviews)
22                     login = True
23                 except WebDriverException :
24                     login = False
25         #creat file csv for reviews
26         with open(name_of_Hotal_file, 'w', newline='') as file:
27             writer = csv.writer(file)
28             writer.writerow
29             (["username" ,"nationality" , "personal_score" , "review_title" , "positive_part" , "negative_part", "date_of_reviews"])

```

Figure 2.17: First part of class reviews()

```

79     get_reveiwis(name_of_Hotal_file)
80     # go to the next page
81     try :
82         next_l = driver.find_element_by_xpath("//a[@class='pagenext']")
83     except NoSuchElementException :
84         driver.refresh()
85         time.sleep(2)
86         try :
87             next_l = driver.find_element_by_xpath("//a[@class='pagenext']")
88             next_l.click()
89         except NoSuchElementException :
90             pass
91     try :
92         while next_l :
93             try :
94                 next_l = driver.find_element_by_xpath("//a[@class='pagenext']")
95                 next_l.click()
96             except NoSuchElementException :
97                 driver.refresh()
98                 time.sleep(2)
99             try :
100                 next_l = driver.find_element_by_xpath("//a[@class='pagenext']")
101                 next_l.click()
102             except NoSuchElementException :
103                 break
104         get_reveiwis(name_of_Hotal_file)
105     except :
106         pass
107     driver.close()

```

Figure 2.18: Second part of class reviews()

The web page shows that just 10 persons commented there, we need to click on the next page to see other reviews. Figure 2.18, line 82 is used to give to the driver the element of the next page and in the line 88 we click on it,

```

29         # function for get the reveiws
30         def get_reveiws(name_of_Hotal_file):
31
32             # to work in block reviews
33             reviews_block = driver.find_elements_by_xpath("//ul[@class='review_list']/li/div[contains(@class,'c-review-block')]")
34             for r in reviews_block :
35                 # get username of The commentator
36                 try:
37                     username = r.find_element_by_xpath("../span[@class='bui-avatar-block__title']").get_attribute("innerHTML")
38                 except NoSuchElementException :
39                     username = ''
40                 # get nationality of The commentator
41                 try:
42                     nationality = r.find_element_by_xpath("../span[@class='bui-avatar-block__subtitle']").get_attribute("innerHTML")
43                     # to delete the part of flag image
44                     get_nationality = nationality.split("</span>")
45                     nationality = get_nationality[1]
46                 except NoSuchElementException :
47                     nationality = ''

```

Figure 2.19: First part of function get_reviews()

```

48         # get personal_score of The commentator
49         try:
50             personal_score = r.find_element_by_xpath("../div[@class='bui-review-score__badge']").get_attribute("innerHTML")
51         except NoSuchElementException :
52             personal_score = ''
53         # get review_title of The commentator
54         try:
55             review_title = r.find_element_by_xpath("../div[@class='bui-grid__column-10']/h3").get_attribute("innerHTML")
56         except NoSuchElementException :
57             review_title = ''
58         # get positive_part of The commentator
59         try:
60             positive_part = r.find_element_by_xpath("../div[@class='c-review_row']/p/span[@class='c-review_body']").get_attribute("inn
61         except NoSuchElementException :
62             positive_part = ''
63         # get negative_part of The commentator
64         try:
65             negative_part = r.find_element_by_xpath("../div[@class='c-review_row_lalala']/p/span[@class='c-review_body']").get_attribu
66         except NoSuchElementException :
67             negative_part = ''
68         try:
69             date_of_reviews = r.find_element_by_xpath("../div[@class='c-review-block__row']/span[@class='c-review-block__date']").get_at
70         except NoSuchElementException :
71             date_of_reviews = ''
72         # append reviews in the csv file
73         with open(name_of_Hotal_file, 'a', newline='') as file:
74             writer = csv.writer(file)
75             writer.writerow([username ,nationality , personal_score , review_title , positive_part , negative_part , date_of_reviews ])
76

```

Figure 2.20: Second part of function get_reviews()

These are all the steps from choosing a city to get the reviews. They are repeated until we complete all the reviews of each hotel in each city. Every country has its own spider with the same code but have a different number of div in the web page of all countries in figure 2.8 line 114. We decided to use this method because the process consume a long time considering that the used calculation material is not very powerful, so we deal with each country apart and cities are processed one by one.

2.6 Data preview

2.6.1 Image of the data collection

These are some reviews of Abdeen Palace Hostel in Egypt

The dataset is available via this link:

https://drive.google.com/file/d/1zCsVU_5EKIQ5iMaPU0NFQZWHVWKdN-n6/view

username	nationality	personal_score	review_title	positive_part	negative_part	date_of_reviews
Mohamed	مصر	10	استثنائي	كل حاجة ممتازة	لا يوجد	تم التقييم: 23 ديسمبر ، 2020
Khaledkhiri	مصر	10	استثنائي	لغرفة جميله جدا .. والناس محترمين وذوق جدا		تم التقييم: 21 ديسمبر ، 2020
Ahmed	مصر	10	استثنائي	فندق ممتاز واستقبال ممتاز وناس محترمه بجد		تم التقييم: 13 ديسمبر ، 2020
Mahmoud	مصر	10	ممتاز وهادئ جدا	تأتي مرة أزور المكان وفعلًا يرتاح جدا فيه ومستر عبداللطيف من الشخصيات المحترمه جدا والمكان جميل جدا	لا يوجد شيء	تم التقييم: 12 ديسمبر ، 2020
Bakr	المملكة العربية السعودية	10	حيث الإقامة	الاقامة ممتازة و مريحه و الخدمة على مستوى عالي	لا يوجد	تم التقييم: 4 ديسمبر ، 2020
Mahmoud	مصر	10	ممتاز جدا	للمعاملة ممتازة جدا المكان نظيف جدا الموقع ممتاز	لا شيء	تم التقييم: 3 ديسمبر ، 2020
العشاري	مصر	10	اتصح به بشدة ممتازااااا	كل شي ابتداء من الطاقم مستر عبداللطيف ومستر اسامه وبقي الموظفين ودودين وبشوشين وفي غاية التعاون	لا شيء على الإطلاق	تم التقييم: 2 ديسمبر ، 2020
Khaledkhiri	مصر			تم إعفاء التقييم لأنه لا يتماشى مع إرشاداتنا.		
Shimaa	مصر	10	استثنائي	تحياتك في بيتك والناس اللي هناك استاذ عبد اللطيف 🙌 و الاستاذ اسامه 🙌 والقمر سمر 🙌🙌🙌 بجد نا	لا شيء . الفندق طريف و عتالي	تم التقييم: 2 أكتوبر ، 2020

Figure 2.21: Some reviews of Abdeen Palace Hostel

2.6.2 Time spent

Given the used material information:

Ram: 8GB

Processor: Intel Core i5 2410M,

Graphics Adapter: AMD Radeon HD 6470M,

Hard Drive: SSD 128GB.

The time to collect the data is huge and the selenium needs a powerful calculation machine to work fast, all this hindrances made collecting of 200,000 reviews take 3 days. We end up collecting 1,604,762 reviews in 30 days.

Title	Number
Number of reviews	1,604,762
Number of hotels	4,533
Max reviews per hotel	75,341
Min reviews per hotel	1

Table 2.1: Statistics on LASHAR dataset

2.6.3 Statistics on LASHAR dataset

2.7 Conclusion

Knowing how a scraper operates and how to make an effective scraper will benefit not only businesses but also individuals looking for specific knowledge. If the economy shifts toward high-tech industries, it's more crucial than ever to get the data you need in a timely and accurate manner. Web scrapers have a lot of promise as a means for accessing the many and unstructured data sources out there. In the next chapter, we will test the data evaluate its quality using Arabic sentiment analysis algorithms.

CHAPTER 3

SENTIMENT ANALYSIS

3.1 Introduction

One of the really common tasks of NLP is identifying and categorizing views conveyed in a piece of text (also known as sentiment analysis). Despite being one of the most widely spoken languages on the planet, Arabic attracts no recognition when it comes to sentiment analysis.

3.2 Data reformatting

Four algorithms were chosen to apply on the dataset during our research on Arabic sentiment analysis algorithms, but the first step is to format the dataset in order to obtain good results. Each person have a "username", "nationality", "personal score", "review title", "positive part", "negative part", and "date of reviews". In our case, the 04 sentiment analysis algorithms need just a negative or positive evaluation. After formatting and deleting the null reviews and the reviews that contain لا يوجد تعليق بهذا الاسم, the latest data looks like figure 3.1.

The dataset is available via this link:

https://drive.google.com/file/d/1iH_s8PY67QxE2Wzve7svTGnDkeqQ6Q56/view .

ID	Feed	Sentiment
1	الموقع علي البحر كان جميل	Positive
2	طاقم الخدمة لم يكن في المستوى المطلوب	Negative
3	الموقع علي البحر كان جميلا جدا ومناسب للأطفال	Positive
4	الطاقم لم يكن في مستوي الخدمة والنظافة مقبولة نوعا ما	Negative
5	الحجرة واسعة ومريحة	Positive
6	الإستغلال التام لكل الخدمات حتى المجانية منها مقارنة بباقي الفنادق حشرات وناموس يغزو الحديقة وناحية المسبح بدون اي معالجة من قبل إدارة الفندق	Negative
7	الموقع جيد	Positive
8	لم يكن بالمستوي خيارات الاكل محدودة والغرف غير مجهزة جيد بدون تلاجة ماء للتكيف لا يعمل لا يوجد علاج من اجل البعوض لا يوجد كاتل شاي او قهوة بالغرفة	Negative

Figure 3.1: Example of the latest data

To edit the dataset, we used the script shown in Figure 3.2

```

1 import csv
2
3 # read raw data rows
4 with open("/home/cherif/projects/booking.com/Tunisia/7000/combined_Tunisia.csv", 'r') as raw:
5     csvRead = csv.DictReader(raw, delimiter=',')
6     # create new dataset file container...
7     with open("grand_nile_tower_reformatteddddddddd.csv", 'w') as new:
8         csvWrite = csv.DictWriter(new, fieldnames=['ID', 'Feed', 'Sentiment'], delimiter=',')
9         csvWrite.writeheader()
10
11         ID = 0
12         for row in csvRead:
13             if row['positive_part'] != '' and row['positive_part'] != "لا يوجد تعليق لهذا التقييم":
14                 ID += 1
15                 new_row_pos = {}
16                 new_row_pos['ID'] = ID
17                 new_row_pos['Feed'] = row['positive_part']
18                 new_row_pos['Sentiment'] = 'Positive'
19                 csvWrite.writerow(new_row_pos)
20
21             if row['negative_part'] != '':
22                 ID += 1
23                 new_row_neg = {}
24                 new_row_neg['ID'] = ID
25                 new_row_neg['Feed'] = row['negative_part']
26                 new_row_neg['Sentiment'] = 'Negative'
27                 csvWrite.writerow(new_row_neg)
28
29 print("done !")
30

```

Figure 3.2: Formatting the dataset

3.3 Text pre-processing

There are some essential steps in the pre-processing of arabic language. These measures include eliminating punctuation, Arabic diacritics (short vowels and other harakahs), elongation, and stopwords (which is available in NLTK corpus).

3.3.1 Library import and Arabic diacritics

Figure 3.3 shows the imported libraries and Arabic diacritics that are needed for this process and figure 3.4 shows the function preprocess that removes punctuations, tashkeel and longation.

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import string
5 import re
6 from nltk.corpus import stopwords
7 from sklearn.model_selection import train_test_split, GridSearchCV
8 from sklearn.feature_extraction.text import TfidfVectorizer
9 from sklearn.pipeline import make_pipeline
10 from sklearn.linear_model import LogisticRegression
11 from sklearn.ensemble import RandomForestClassifier
12 from sklearn.naive_bayes import MultinomialNB
13 from sklearn.svm import SVC
14 from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
15
16 data = pd.read_csv(r"/home/cherif/projects/booking.com/BookingComAllReviews.csv")
17

```

Figure 3.3: Library import and arabic diacritics

3.3.2 Function preprocess text

```

43 def preprocess(text):
44     """
45     text is an arabic string input
46
47     the preprocessed text is returned
48     """
49
50     # remove punctuations
51     translator = str.maketrans('', '', punctuation)
52     text = text.translate(translator)
53
54     # remove Tashkeel
55     text = re.sub(arabic_diacritics, '', text)
56
57     # remove longation
58     text = re.sub("([|])", "", text)
59     text = re.sub("و", "و", text)
60     text = re.sub("ة", "ة", text)
61     text = re.sub("ة", "ة", text)
62     text = re.sub("ة", "ة", text)
63     text = re.sub("ة", "ة", text)
64
65     text = ' '.join(word for word in text.split() if word not in stop_words)
66
67     return text
68
69 data['Feed'] = data['Feed'].apply(preprocess)
70 print(data)

```

Figure 3.4: Preprocess function

3.3.3 Example of pre-processing text

ID	Feed	Sentiment
0 1	اريد فيها جامعات اكثر من عمان ... وفيها قد عم	Positive
1 2	الحلو انكم بتحكوا على اساس انو الاردن ما فيه	Negative
2 3	كله رائع بجد ربنا يكرمك	Positive
3 4	لسانك قذر يا قمامة	Negative
4 5	اتقوا الله فينا بكفي رفع اسعار \$\$\$ الرواتب با	Negative

Figure 3.5: Text before

ID	Feed	Sentiment
0 1	اريد جامعات اكثر عمان وفيها ونص لعيه الم	Positive
1 2	الحلو انكم بتحكوا على اساس انو الاردن فساد سرقات	Negative
2 3	كله رائع بجد ربنا يكرمك	Positive
3 4	لسانك قذر قمامة	Negative
4 5	اتقوا الله فينا بكفي رفع اسعار الرواتب بالحضيض	Negative

Figure 3.6: Text after

3.3.4 Natural Language Toolkit

NLTK is “A great platform for teaching and working in computational linguistics using Python,” as well as “an excellent library to play with natural language.”.[33] NLTK is a popular Python platform for dealing with human language data. Text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning are used, as well as wrappers for industrial-strength. [32]

3.4 Google collaboratory (colab)

Colab runs in the cloud, which provides a Jupiter notebook environment and allows the creation of notebooks with Python. Google colab has a free Jupiter and a pro one the difference between them is shown in Table 3.1:

	Price	GPU	Runtime	Memory
Colab	Free	K80	Up to 12 hours	12GB
Colab pro	Paid	T4 & P100	Up to 24 hours	25GB with high memory VMs

Table 3.1: comparison of Colab and Colab pro[33]

Due to the lack of a high-performance computational machine, taking into consideration that the university server is not an option because it doesn’t have a GPU, we had to buy access to google colab pro server.

3.5 Arabic sentiment analysis algorithms

In this section, we will consider the logistic regression, random forest classifier naive Bayes classifier (Multinomial), and support Vector machine algorithms.

Note: Even with the use of google colab the process takes a long time, so we applied the algorithms on a chunk of 20k of the data.

3.5.1 Logistic Regression

The classification algorithm logistic regression is widely used. It’s easy to implement and can be used as a baseline algorithm for classification tasks. A Pipeline class in scikit-Learn, which incorporates vectorization, transformation, grid-search, and classification,

is used to render the code shorter. The code of the logistic regression algorithm appears in figure 3.7

```

19 # splitting the data into target and feature
20 feature = data.Feed
21 target = data.Sentiment
22 # splitting into train and tests
23 X_train, X_test, Y_train, Y_test = train_test_split(feature, target, test_size=.2, random_state=100)
24 # ##### Logistic Regression #####
25 def Logistic_Regression():
26     # make pipeline
27     pipe = make_pipeline(TfidfVectorizer(), LogisticRegression())
28     # make param grid
29     param_grid = {'logisticregression__C': [0.01, 0.1, 1, 10, 100]}
30
31     # create and fit the model
32     model = GridSearchCV(pipe, param_grid, cv=5)
33     model.fit(X_train, Y_train)
34
35     # make prediction and print accuracy
36     prediction = model.predict(X_test)
37     print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
38     print(classification_report(Y_test, prediction))
39

```

Figure 3.7: Function of logistic regression

Figure 3.8 shows the result of this algorithm, which is very fast and does not take a long time to execute. Time spent for training is 5s. The result appears in figure 3.8

```

Accuracy score is 0.88
              precision    recall  f1-score   support

   Negative      0.87      0.88      0.88      1951
   Positive      0.89      0.87      0.88      2049

 accuracy              0.88              4000
 macro avg      0.88      0.88      0.88      4000
 weighted avg    0.88      0.88      0.88      4000

0.514032260576884

```

Figure 3.8: Result of logistic regression

A score of 0.88 accuracy was obtained.

3.5.2 Random Forest Classifier

Random forests, also called random decision forests, works by training a large number of decision trees and then outputting the class that is the mode of the classes (classification) or the mean/average predictor (regression) of the individual trees which makes it a learning system for classification. Figure 3.9 shows the code of the random forest classifier algorithm.

```

40 # ##### Random Forest Classifier #####
41 def Random_Forest_Classifier():
42
43     pipe = make_pipeline(TfidfVectorizer(), RandomForestClassifier())
44
45     param_grid = {'randomforestclassifier__n_estimators':[10, 100, 1000], 'randomforestclassifier__max_features':['sqrt', 'log2']}
46
47     rf_model = GridSearchCV(pipe, param_grid, cv=5)
48     rf_model.fit(X_train, Y_train)
49
50     prediction = rf_model.predict(X_test)
51     print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
52     print(classification_report(Y_test, prediction))

```

Figure 3.9: Function of random forest classifier

The training phase of this algorithm takes a long time to complete. Time spent for training is: 57 min. The result appears in figure 3.10

```

Accuracy score is 0.88
precision    recall  f1-score   support

Negative     0.87     0.89     0.88     1951
Positive     0.89     0.87     0.88     2049

accuracy          0.88     4000
macro avg         0.88     0.88     0.88     4000
weighted avg      0.88     0.88     0.88     4000

57.95343989133835

```

Figure 3.10: Result of random forest classifier

A score of 0.88 accuracy was obtained.

3.5.3 Naive Bayes Classifier (Multinomial)

Many other types of classifiers use the expensive iterative approximation, but Naïve Bayes classifiers can do Maximum-likelihood training by evaluating a closed-form expression, which takes linear time. The number of parameters used by Naïve Bayes classifiers is linear regarding the number of variables (features/predictors) in a learning problem, making them highly scalable. The naive Bayes classifiers are a basic "probabilistic classifier" based on Bayes' theorem and strict (naïve) independence assumptions between the features. Code of the naive Bayes classifier algorithm appears in figure 3.11

Time spent for training is: 2s. The result appears in figure 3.12

```

54 # ##### Naive Bayes Classifier (Multinomial) #####
55 def Naive_Bayes_Classifier():
56     pipe = make_pipeline(TfidfVectorizer(), MultinomialNB())
57     pipe.fit(X_train, Y_train)
58     prediction = pipe.predict(X_test)
59     print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
60     print(classification_report(Y_test, prediction))
61

```

Figure 3.11: Function of naive Bayes classifier

A score of 0.88 accuracy was obtained.

Accuracy score is 0.88					
	precision	recall	f1-score	support	
Negative	0.88	0.87	0.88	1951	
Positive	0.88	0.89	0.88	2049	
accuracy			0.88	4000	
macro avg	0.88	0.88	0.88	4000	
weighted avg	0.88	0.88	0.88	4000	
0.06505133310953776					

Figure 3.12: Result of naive Bayes classifier

3.5.4 Support Vector Machine

Support-vector machines, which are based on mathematical learning systems or VC theory suggested by Vapnik (1982, 1995) and Chervonenkis, are one of the most reliable prediction methods (1974). Supporting vector machines use an educational data analysis algorithm for classification and regression analysis, categorized into supervised educational models. SVM was developed in ATT Bell Laboratories. Figure 3.13 shows code of the SVM algorithm

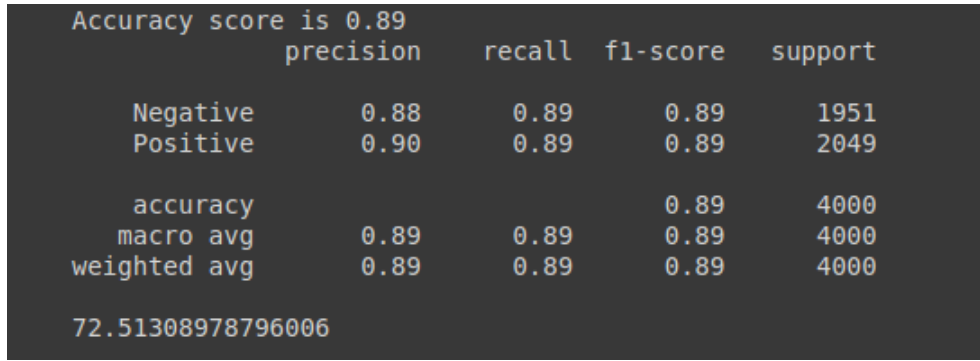
```

62 # ##### Support Vector Machine #####
63 def Support_Vector_Machine():
64     pipe = make_pipeline(TfidfVectorizer(), SVC())
65     param_grid = {'svc__kernel': ['rbf', 'linear', 'poly'], 'svc__gamma': [0.1, 1, 10, 100], 'svc__C': [0.1, 1, 10, 100]}
66
67     svc_model = GridSearchCV(pipe, param_grid, cv=3)
68     svc_model.fit(X_train, Y_train)
69
70     prediction = svc_model.predict(X_test)
71     print(f"Accuracy score is {accuracy_score(Y_test, prediction):.2f}")
72     print(classification_report(Y_test, prediction))
73

```

Figure 3.13: Function of support vector machine

Time spent for training is: 72 min. The result appears in figure 3.14



```

Accuracy score is 0.89
              precision    recall  f1-score   support

   Negative      0.88      0.89      0.89     1951
   Positive      0.90      0.89      0.89     2049

 accuracy
macro avg      0.89      0.89      0.89     4000
weighted avg    0.89      0.89      0.89     4000

72.51308978796006

```

Figure 3.14: Result of support vector machine

A score of 0.89 accuracy was obtained.

3.5.5 Evaluation of the used algorithms

		Precision	recall	F1-score	Time	Accuracy score
LR	Negative	0.87	0.88	0.88	5s	0.88
	Positive	0.89	0.87	0.88		
RFC	Negative	0.87	0.89	0.88	57m	0.88
	Positive	0.89	0.87	0.88		
NBC	Negative	0.88	0.87	0.88	2s	0.88
	Positive	0.88	0.89	0.88		
SVM	Negative	0.88	0.89	0.89	72m	0.89
	Positive	0.90	0.89	0.89		

Table 3.2: Comparison between the algorithms

We used four well-known sentiment classifiers to examine the dataset's validity and efficiency. The best results came from SVM classifiers. the recorded accuracy ranges from 88 to 89 percent for polarity classification. Our primary commitment is to make this benchmark dataset set available and open to the Arabic language research community. We believe that this dataset will be useful, and will spark further studies in the Arabic sentiment analysis and related issues.

3.6 Comparison between LASHAR and the different datasets

	LABR	ASTD	ALC	HARD	ATSAD	AVTCD	LASHAR
Size of the dataset	63,000 Arabic reviews	10,000 Arabic tweets	240,000 Words	490,587 reviews	36,868 Tweets	557,576 Tweets	1604,762 Review

3.7 Conclusion

In this chapter, we reformatted the dataset for learning purposes, preprocessed the text, and normalized it. we presented how we apply four sentiment analysis algorithms different on the dataset. The result of sentiment analysis algorithms validates and indicates the quality and strength of this dataset, as well as proof that the data gathering procedure was based on a study rather than randomly, making it an effective and helpful addition to sentiment analysis science.

GENERAL CONCLUSION

There are multiple platforms and websites that serve as a good resource of Arabic reviews. We chose “booking.com” of all others because it provides us with a huge advantage which is represented in the already classified reviews (by providing a rating score for each and dividing the positive part from the negative one). The Corpus notation task is a costly one since it requires many human resources and a greater time limit as the data grows. In this thesis, we provided an exhaustive study about web scraping and how to use it to collect linguistic data. After that, we reformatted the dataset for learning purposes, preprocessed the text, and normalized it. Finally, we presented how useful this data is by applying some SA algorithms on it where the results were so promising. We believe that there is one major positive side effect of this work which needs to be highlighted. Providing SA pre-trained models will help us get through the notation problem with websites that don’t provide this feature. With that challenge dealt with, reaching higher targets like one BILLION Arabic notated reviews is just a matter of time.

REFERENCES

- [1] <https://www.ibm.com/cloud/learn/natural-language-processing>. May 2021.
- [2] Nielsen, M. A. Neural networks and deep learning, volume 2018. Determination press San Francisco, CA.(2015).
- [3] Hutchins, J. "The history of machine translation in a nutshell".(2005).
- [4]https://www.softwaretestinghelp.com/what-is-artificial-intelligence/5_Natural_Language_Processing . May 2021.
- [5] John A. Bullinaria . IAI : The Roots, Goals and Sub-fields of AI . 2005
- [6] Elizabeth D. Liddy. Natural Language Processing . 2001 .
- [8] WEB SITE <http://www.contrib.andrew.cmu.edu/~dyafei/NLP.html>. May 2021.
- [9] N. Boukhatem, "The Arabic natural language processing: Introduction and challenges," Int. J. English Lang. Transl. Stud., vol. 2, no. 3, pp. 106–112, 2014.
- [10] A. A. Al-Ajlan, H. S. Al-Khalifa, and A. S. Al-Salman, "Towards the development of an automatic readability measurements for Arabic language," in Proc. 3rd Int. Conf. Digit. Media, Nov. 2008, pp. 506–511.
- [11] A. Farghaly and K. Shaalan, "Arabic natural language processing: Challenges and solutions," ACM Trans. Asian Lang. Inf. Process., vol. 8, no. 4, p. 14, 2009.
- [12] (Bassam Hammo,Hani Abu-Salem, Steven Lytinen).QARAB: A Question Answering System to Support the Arabic Language.2002.
- [13] Author : Naaima Boudad , Rdouan Faizi , Rachid Oulad , Haj Thami , Raddouane Chiheb .Sentiment analysis in Arabic: A review of the literature. 2017.
- [14] <https://byjus.com/maths/data-sets/>. June 2021

- [15] <https://labeledyourdata.com/articles/what-is-dataset-in-machine-learning/>. May 2021.
- [16] <https://www.definedcrowd.com/blog/the-challenge-of-building-corpus-for-nlp-libraries/>. May 2021.
- [17] Mohamed Aly and Amir Atiya .LABR: A Large Scale Arabic Book Reviews Dataset. 2013.
- [18] Mahmoud Nabil, Mohamed Aly, Amir F. Atiya. ASTD: Arabic Sentiment Tweets Dataset. 2015.
- [19] Abdullah ALFAIFI, Eric ATWELL, Hedaya IBRAHEEM. Arabic Learner Corpus (ALC). 2014 .
- [20] Ashraf Elnagar, Yasmin S. Khalifa and Anas Einea. Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications.2016.
- [21] Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, Simon Dobnik, Richard Johansson. An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training.2020 .
- [22] Ayman Alhelbawy, Udo Kruschwitz, Massimo Poesio. Towards a corpus of violence acts in Arabic social media. 2016 .
- [23] Mahto D, Singh L. A Dive into Web Scraper World. Computing for Sustainable Global Development (INDIACom).2016.
- [24] Website: <https://docs.scrapy.org/en/0.22/topics/architecture.html> . May 2021
- [25] <https://developer.mozilla.org/en-US/docs/Web/XPath> . May 2021
- [26] https://developer.mozilla.org/enUS/docs/Learn/CSS/Introduction_to_CSS/Selectors. May 2021
- [27] <https://docs.scrapy.org/en/latest/> . May 2021
- [28] <https://webbrowsersintroduction.com/> . May 2021
- [29] <https://splash.readthedocs.io/en/stable/> . May 2021
- [30] Lawson R. Web Scraping With Python. 2015.
- [31] Website:<https://www.javatpoint.com/selenium-webdriver> . May 2021.
- [32]Steven Bird Edward Loper. NLTK: The Natural Language Toolkit. 2004.
- [33]Website : <https://buomsoo-kim.github.io/colab/2020/03/15/Google-newly-launches-colab-pro.md/>. May 2021