Improving the Accessibility of Arabic Electronic Theses and Dissertations (ETDs) with Metadata and Classification

Eman H. Abdelrahman

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Computer Science and Applications

Osman Balci, Co-chair
Edward Fox, Co-chair
Reza Barkhi

November 29, 2021
Blacksburg, Virginia

Keywords: Digital Libraries, Arabic Electronic Theses and Dissertations (ETDs), Automatic Classification, Machine Learning, Deep Learning, Pretrained Language Models

Improving the Accessibility of Arabic Electronic Theses and Dissertations (ETDs) with Metadata and Classification

Eman H. Abdelrahman

(ABSTRACT)

Much research work has been done to extract data from scientific papers, journals, and articles. However, Electronic Theses and Dissertations (ETDs) remain an unexplored genre of data in the research fields of natural language processing and machine learning. Moreover, much of the related research involved data that is in the English language. Arabic data such as news and tweets have begun to receive some attention in the past decade. However, Arabic ETDs remain an untapped source of data despite the vast number of benefits to students and future generations of scholars. Some ways of improving the browsability and accessibility of data include data annotation, indexing, parsing, translation, and classification. Classification is essential for the searchability and management of data, which can be manual or automated. The latter is beneficial when handling growing volumes of data. There are two main roadblocks to performing automatic subject classification on Arabic ETDs. The first is the unavailability of a public corpus of Arabic ETDs. The second is the Arabic language's linguistic complexity, especially in academic documents. This research presents the Otrouha project, which aims at building a corpus of key metadata of Arabic ETDs as well as providing a methodology for their automatic subject classification. The first goal is aided by collecting data from the AskZad Digital Library. The second goal is achieved by exploring different machine learning and deep learning techniques. The experiments' results show that deep learning using pretrained language models gave the highest classification performance, indicating that language models significantly contribute to natural language understanding.

Improving the Accessibility of Arabic Electronic Theses and Dissertations (ETDs) with Metadata and Classification

Eman H. Abdelrahman

(GENERAL AUDIENCE ABSTRACT)

An Electronic Thesis or Dissertation (ETD) is an openly-accessible electronic version of a graduate student's research thesis or dissertation. It documents their main research effort that has taken place and becomes available in the University Library instead of a paper copy. Over time, collections of ETDs have been gathered and made available online through different digital libraries. ETDs are a valuable source of information for scholars and researchers, as well as librarians. With the digitalization move in most Middle Eastern Universities, the need to make Arabic ETDs more accessible significantly increases as their numbers increase. One of the ways to improve their accessibility and searchability is through providing automatic classification instead of manual classification. This thesis project focuses on building a corpus of metadata of Arabic ETDs and building a framework for their automatic subject classification. This is expected to pave the way for more exploratory research on this valuable genre of data.

Dedication

To my beloved father Hussein Kamel Abdelrahman (RIP) and mother Kamilia Basiouny, my brothers Abdelrahman and Omar, and my husband Karim Youssef. To my precious baby-in-heaven Adam. To my beloved parents-in-law Yasser and Nancy. To my loving aunt Magda, her husband Lotfy Soliman, and my cousins Mona, Amira (RIP), Iman, Hoda, Marwa, and Heba, who supported me wholeheartedly. To my friends who helped me during tough times. I am forever grateful for all of them.

Acknowledgments

I would like to greatly thank my advisors and mentors, Prof. Edward Fox and Prof. Osman Balci, for their continuous support and guidance. They have been keen to put me on the right track and have been patient and understanding whenever needed. Thanks also go to Dr. Reza Barkhi for being part of my committee and always being willing to help.
Thanks go to the AskZad Digital Library, which is part of the Saudi Digital Library, for providing access to data.

Special thanks to Fatimah Alotaibi for her support all the time. I would also like to thank Palakh Jude for her assistance. I would like to thank Mohamed Magdy Farag for his assistance in the early stages of this research work.

I would also like to thank Josh Vinson from the Virginia Tech Writing Center for his consistent assistance in writing my thesis.

This project was made possible in part by the Institute of Museum and Library Services grant LG-37-19-0078-19.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic **T**heses and **D**issertations (ETDs) constitute one of the genres of data that is rich in information but has been unexplored by researchers until recently. This research work focuses on Arabic ETDs, even less explored.

This project's first goal is to build a corpus of metadata of Arabic ETDs by searching the AskZad Digital Library, which is a component of the Saudi Digital Library (SDL). The dataset collected consists of key metadata of Arabic ETDs such as abstracts, titles, and keywords. Subsequent preprocessing included stopword removal using NLTK [34], and word lemmatization using the Farasa API [2]. After that, different machine learning and deep learning techniques have been applied to provide efficient automatic classification.

The second goal is to provide a methodology for the automatic subject classification of Arabic ETDs. Through subject classification, it is expected to make Arabic ETDs more accessible and better browseable. This will also facilitate organizing and managing Arabic ETDs in digital libraries since their number is growing significantly. This is expected to encourage more research to take place on Arabic ETDs. In addition, this research work could be further extended to provide machine learning as a service (MLaaS), which would also benefit researchers and digital libraries.

1.1 Background

While some Arabic text data has been studied in machine learning and natural language processing, most of it consists of news articles and tweets, which are readily available to the general populace. However, Arabic **E**lectronic **T**heses and **D**issertations (ETDs) have received less to no attention, although they provide numerous benefits to students, researchers of all backgrounds, scholars, and universities in general.

This is mainly due to two main roadblocks: (1) the unavailability of a corpus of Arabic ETDs and (2) the linguistic complexity of the formal Arabic language. The latter places a heavy strain on the preprocessing of the data and creates a challenge to apply common machine learning and deep learning techniques effectively.

1.2 Problem Statement

ETDs provide many benefits to researchers and scholars. Therefore, research work has been done to archive them and make them accessible through digital libraries.

Several universities in the Middle East have started to require an Arabic version of students' ETDs or at least an Arabic version of their abstracts and other metadata. Accordingly, the number of Arabic ETDs is growing gradually. However, to the best of our knowledge, Arabic ETDs have neither been explored nor made publicly accessible as have other data genres. There are few available structured corpora of Arabic ETDs available for researchers. Accordingly, the first goal of this thesis project is to build a corpus of data that consists of the key metadata of Arabic ETDs.

One of the ways to improve the accessibility of digital text data is through automatic subject classification, especially when works are not cataloged by librarians or domain experts.

Therefore, the second goal of this project is to research different approaches to automatic classification of Arabic text. We aim to provide a methodology for the automatic subject classification of Arabic ETDs based on their metadata. This is expected to make the process of digitalizing a considerable amount of ETDs, especially those that are not born digital, more efficient, and to make them more accessible and searchable. We hope that more such research will take place on Arabic ETDs.

1.3 Motivation

Extensive work to digitally store ETDs has enabled researchers to find what their peers have previously worked on and use their valuable information. With the number of Arabic ETDs gradually increasing, it becomes more challenging to efficiently store them and make them digitally accessible and browseable. This is where automatic text classification comes in. The lack of an available public corpus of ETDs made them largely untapped in the realm of machine learning and natural language processing.

To overcome this challenge and encourage more research to use this unexplored genre of data (Arabic ETDs), a structured corpus of key metadata of Arabic ETDs is built as part of the bigger goal of this research work.

1.4 Hypotheses

This research project is based on five hypotheses. The first is: machine learning and natural language processing techniques can accurately automatically classify Arabic ETDs, based on their metadata. The second is: regarding selection of a suitable classification method, we hypothesize that: automatic classification using deep learning would not be suitable if the

size of the corpus is not large.

However, building the corpus gradually as we did, it is not known how much data would be enough for suitable classifiers to be built. So, addressing how much data is needed, the third hypothesis is: the more documents the corpus contain, the more accurate the automatic subject classification becomes. Fourth, regarding what data is needed, we hypothesize that: if more data is added to each dataset document, the accuracy will increase of the automatic subject classification. For example, regarding classifying by subject category, using abstracts only might do less well than when abstracts, titles, and keywords are used together.

Fifth, regarding how this works with Arabic vs. other languages, we hypothesize that: the language itself can contribute to the performance of the classification model. For example, automatically classifying the Arabic version of the metadata would result in different performance than classifying the English version of the metadata.

1.5 Research Questions

This thesis will attempt to answer the following research questions:

- How can we build a corpus of Arabic ETDs' metadata to be used for research?
- Can we get satisfactory classification performance using only key metadata?
- How can machine learning techniques be used to automatically classify Arabic ETDs based on their metadata i.e., abstract, title, and keywords?
- Will deep learning techniques improve automatic classification of Arabic ETDs based on their metadata?

1.6 Research Contributions

This research work makes the following contributions:

- This work provides a corpus of metadata of Arabic ETDs, collected from the AskZad Digital Library. This will make research on Arabic ETDs more possible in the future.
- It outlines a methodology for the automatic subject classification of Arabic ETDs based on their metadata using the AskZad categorization system.
- It explains the impact of linguistic differences on automatic classification and how to overcome this challenge.
- It explains the benefit of using pretrained language models to overcome the lack of large-scale data for training deep learning models for Arabic ETDs.

1.7 Outline of the Thesis

- Chapter 1 includes the introduction, including general background, problem statement, motivation, hypotheses, research questions, and research contributions.
- Chapter 2 discusses the relevant literature for text preprocessing, classical machine learning approaches for automatic classification of Arabic text, and deep learning approaches.
- Chapter 3 presents the approach adopted in this work, as well as a detailed description of the dataset being built and the developed classification model.
- Chapter 4 presents experimental setup and results (supervised ML and DL) and a discussion of the results.

- Chapter 5 provides conclusions of the research.
- Chapter 6 suggests potential future work.

Chapter 2

Review of Literature

This chapter presents related work in the Arabic text classification area that has taken place on different genres of data such as tweets and news. Research that has used machine learning and deep learning for Arabic text classification has helped make a baseline for this project's experiments.

2.1 Data Preprocessing

Preprocessing is an important task and critical step in **N**atural **L**anguage **P**rocessing (NLP) and **I**nformation **R**etrieval (IR) [25]. Different preprocessing techniques, such as text normalization, stopword removal, stemming, and lemmatization, can be used. Since Arabic does not have uppercase and lowercase versions of each alphabetic character, that type of text normalization is not needed. The text prepossessing techniques considered are the following:

- Stopword Removal: Stopwords are words that are considered meaningless or were too frequent. Examples of stopwords in English are "a", "the", and "are". Stopword removal is essential in reducing dimensionality and in focusing on essential words only [45]. This is applied to text in any language.

- Stemming: It removes or stems the last few characters of a word, often leading to incorrect meanings and spelling [24].

- Lemmatization: It considers the context and converts the word to its meaningful base form (dictionary form), the lemma. Sometimes, the same word can have multiple different lemmas, depending on the context [24].

The Arabic language has a very rich morphology, and the structure of words is complex compared to English words. For example, a root of a verb can have a prefix, infix, and suffix. Prefixes refer to both the time and the tense of the verb. The suffixes contain the gender of the participant in the verb, and reflect singularity or plurality [15]. This richness increases the dimensionality of word vectors. Therefore, it is needed to get the root of each word in the raw dataset. After studying the impact of stemming in **N**atural **L**anguage **P**rocessing (NLP) tasks for Arabic, it is found that lemmatization shows more efficacy than stemming, particularly in text summarization [17], text indexing [21], and text classification [30]. Figure 2.1 gives an example of the complex structure of Arabic words [8], and Figure 2.2 gives an example of the difference between stemming and lemmatization [27].

2.2 Text Classification

Text classification is considered one of the major research topics in Machine Learning (ML) and Natural Language Processing (NLP). Its purpose is categorizing and labeling text into predefined classes or categories based on its content [3]. Text classification tasks include several applications such as sentiment analysis, language detection, topic labeling, and intent detection. The amount of textual data available online is growing rapidly in the digital age, making text classification essential to access, browse, and preserve that data efficiently. Applying this to Arabic text classification is a challenging research area. This is due to Arabic's complex linguistic characteristics, limited studies on it, and limited open access data available. Therefore, most previous research on text classification has been conducted

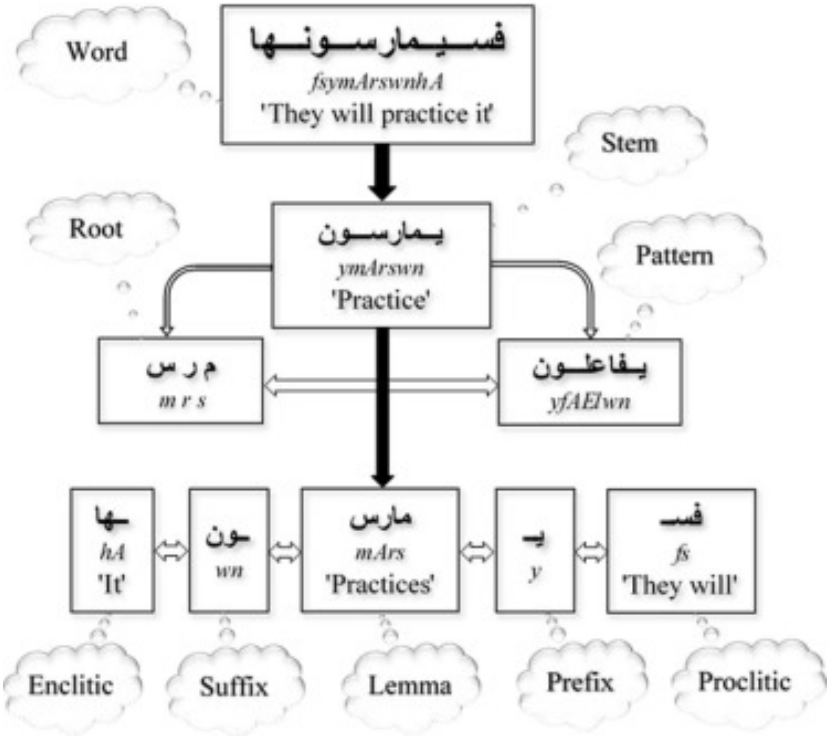

Figure 2.1: Example of Complex Structure of Arabic Words [8].

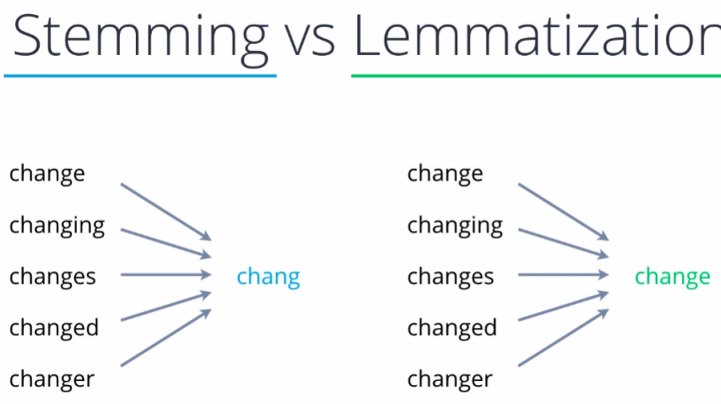

Figure 2.2: Stemming vs. Lemmatization [27]

on English datasets [1]. To the best of our knowledge, there has been no previous research on automatic text classification of Arabic Electronic Theses and Dissertations (ETDs). All of the earlier studies are on different genres of data, such as tweets and news articles [15] [19] [26] [31] [44].

There are multiple types of classification problems including, but not limited to, multiclass classification, multilabel classification, and binary classification. Multiclass classification is the problem where each instance will be assigned to only one of three or more output classes. Multilabel classification is the problem where each instance can be assigned to multiple output classes. On the other hand, binary classification is where each instance will be assigned to only one of two output classes [10].

The difference between binary and multiclass classification problems is shown in Figure 2.3, while Figure 2.4 shows the difference between multiclass and multilabel classification problems. Multiclass problems can make use of binary classification. This can be done by training multiple binary classifiers, one for each class. This is called the one-vs-rest approach [10] and is shown in Figure 2.5. Instead of training a single classifier, one can combine each of the classifiers' binary outputs to generate multiclass outputs. The following subsection demonstrates different classifiers from related work.

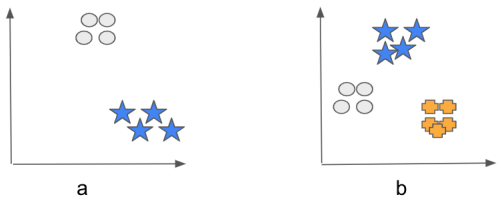

Figure 2.3: Binary (a) vs. Multiclass (b) Classification Problems.

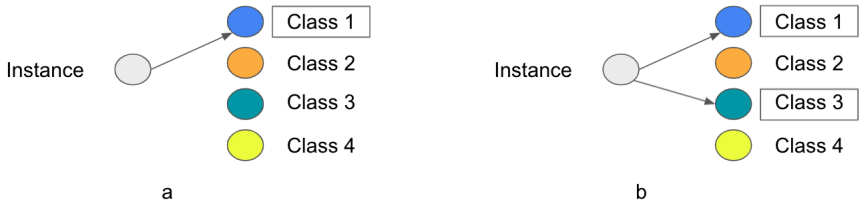

Figure 2.4: Multiclass (a) vs. Multilabel (b) Text Classification Problems.

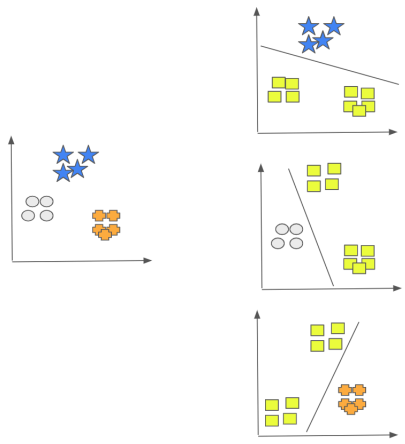

Figure 2.5: One-vs-All Classification Problem (3 Classes).

2.2.1 Classical Machine Learning Approaches

Different classification models have been used in research on Arabic text classification, such as Decision Trees [41], Naïve Bayes [47], and Support Vector Machines [35].

In [15], preprocessing techniques include stopword removal, root extraction, and stemming for dimensionality reduction. A performance comparison has been made between different classification techniques such as Naïve Bayes [39], k-Nearest Neighbor (k-NN) [18], and distance-based classification methods [14], using an Arabic dataset of 1,000 documents of magazines and newspapers. To compare the accuracy between these classifiers, the authors used error rate, recall, precision, and fallout. The experiment showed that Naïve Bayes outperformed the k-NN and distance-based methods.

In [3], an automated tool for Arabic text classification has been presented. One goal of that paper was to build a representative training dataset covering different types of text categories, which can be used for further research. This was done by collecting seven different dataset genres that contained 17,658 text documents with more than 11,500,000 words as their corpus. Their corpus included newswire stories, discussion forums, Arabic poems, etc. Their second goal was to make a performance comparison between the SVM and C5.0 algorithms on the dataset. In general, the study found that the C5.0 algorithm outperformed the SVM algorithm by approximately 10%.

The work in [19] includes using SVM, Naïve Bayes, k-NN, and Rocchio classifiers to categorize Arabic text. Their dataset consists of newspapers. To test the classifiers, they conducted two experiments. In the first, they split the data into training and test sets, while the second experiment involved the leave-one-out testing method. The study found that the Rocchio classifier gave better results when the size of the feature set was small, while SVM outperformed the other classifiers when the size of the feature set was large.

The behavior of n-gram frequency statistics for Arabic text classification was studied in [26]. The dataset used was an online Arabic newspaper. The author employed the Manhattan distance for dissimilarity measure and Dice's measure of similarity along with an n-gram frequency statistics technique. The experiment showed better results for the latter.

An Arabic document categorization tool was developed in [31] using the Naïve Bayes algorithm. For this study, non-vocalized Arabic web documents were classified according to five predefined categories using 300 web documents of news. A cross-validation experiment using 2,000 terms/roots showed an average accuracy over all categories of 68.78%, where the best categorization performance per category was 92.8%.

In contrast to the previous literature, [44] experimented on a large Arabic newswire corpus without preprocessing. The authors posited that statistical methods are powerful for Arabic text classification and clustering (maximum entropy). The results show 89.5%, 31.5%, and 46.61% for recall, precision, and F-measure, respectively. This would generally be viewed as unacceptable, possibly because of the lack of morphological analysis.

As stated in [16], developing a classifier for Arabic text is difficult due to the complexity of Arabic morphological analysis. The Arabic language has high inflectional and derivational morphology, which makes NLP tasks nontrivial. They used the maximum entropy framework to build a system (ArabCat) that works as a classifier for Arabic documents. Their dataset was collected from Arabic websites. Their results show that ArabCat gave 80.48%, 80.34%, and 80.41% for recall, precision, and F-measure, respectively, while the Sakhr categorizer [43] shows 73.78%, 47.35%, and 57.68%, respectively.

2.2.2 Deep Learning

In the area of deep learning, there are several research studies on Arabic text classification. For example, [22] pre-processed their data by transforming all text into a Term-Document matrix (T-D matrix) before remodeling it based on Singular-Value Decomposition (SVD) to reduce its dimensionality. The latter became the input for a three-layer feed-forward neural network with a hyperbolic tangent (tanh) activation function in the hidden layer, followed by a linear output layer. Their evaluation presented best average recall values of 50-53% and best average precision values of 53-55%.

Alternatively, another study [23] consisted of using a combination of Markov clustering with a deep learning approach where documents are preprocessed by tokenizing them, segmenting them into different words, and extracting the TF-IDF weighted root words (which are used as features) counts. Clustering is then performed on the features using fuzzy-c-means and a Markov model before training a deep belief network for each cluster using restricted Boltzmann machines. The author claims this approach improves classification accuracy and feature extraction. The evaluation stage included 10-fold cross-validation on 12,000 news documents, resulting in an F-measure of 91.02%.

Boukil et al. [9] employed a Convolutional Neural Network (CNN) trained on Arabic text found online. They used a web-crawler to build a corpus of 319 million Arabic words and applied the TF-IDF technique. Next, they trained the CNN model using stochastic gradient descent (backpropagation) as well as using a learning rate of 0.001 and Keras' dropout function to enable the model to converge better. They found the two important features in classifying Arabic words were filter sizes and feature maps. The highest accuracy was 92.94% with 111,000 words.

[11] involves a convolutional neural network combined with a differential evolution algorithm

to perform sentiment classification on Arabic text. Their contributions include building and training different CNN architectures with variable numbers of parallel convolution layers, integrating two different mutation strategies to improve the exploration and exploitation abilities of the differential evolution algorithm, and using two different fitness evaluation techniques to assess the generalization of the CNN.

None of these research efforts have focused on Arabic ETDs, which is the core contribution of this research. Instead, the majority of prior research has been applied to other forms of Arabic text such as news and tweets. Furthermore, most of this classification work has relied on stemming the Arabic words in their preprocessing phase, whereas lemmatization has been performed in our work.

One of the ways to better understand languages and relationships between words is through language models. Language modeling (LM) uses various statistical and probabilistic techniques to determine the probability of a given sequence of words occurring in a sentence. Language models analyze bodies of text data to provide a basis for their word predictions [32].

Using Pretrained Language Models (PLMs) has become a dominant approach, improving performance on many NLP tasks through transfer learning. A PLM is a language model that has been trained with a large dataset to *learn* and *understand* the language, to be used for other NLP tasks [7].

Transfer learning in NLP uses pretrained deep learning models on different NLP tasks in that language on which these models will be employed. A pretrained model can be fine-tuned by being trained on a task-specific dataset to create the final model that is capable of performing the required task. Recently, the field of Natural Language Processing (NLP) has witnessed the emergence of several transfer learning methods and architectures, which

significantly improved upon the state-of-the-art on a wide range of NLP tasks [40]. Figure 2.6 shows the workflow of transfer learning.

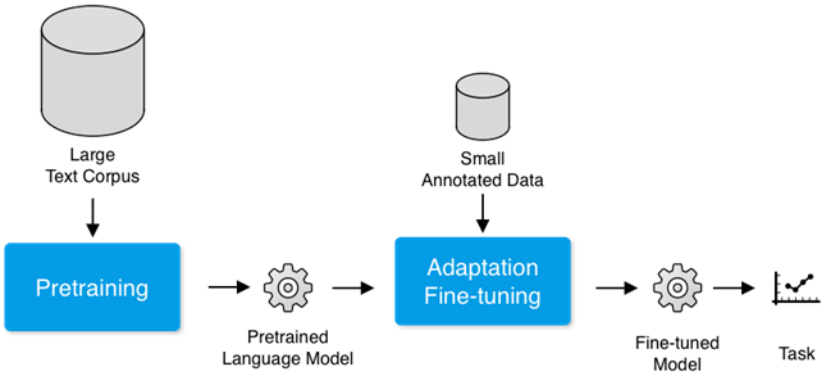

Figure 2.6: Transfer Learning Workflow [20].

Using pretrained language models such as BERT has improved performance in different NLP downstream tasks, such as sentiment analysis, text classification, and machine translation [46]. In addition, using them offloads the burden of training from scratch, which is very costly.

One of the state-of-the-art models is BERT (Bidirectional Encoder Representations from Transformers) [13]. BERT's key technical innovation is applying the bidirectional training of a transformer, a popular attention model, to language modeling. This is in contrast to previous efforts, which looked at a text sequence either from left to right, or combined left-to-right and right-to-left training. A bidirectionally trained language model can have a more profound sense of language context and flow than single-direction language models. Therefore, the following BERT-based models were explored and fine-tuned using our ETD data.

SciBERT [7] (for English data): is a pretrained language model based on BERT [13] to address the lack of high-quality, large-scale labeled scientific data. It leverages unsupervised

pretraining on a large multi-domain corpus of scientific publications to improve the performance on downstream scientific NLP tasks. SciBERT has its own vocabulary (SCIVOCAB) that's built to best match the training corpus. The authors trained cased and uncased versions. There are four versions of the SciBERT model based on: (i) cased or uncased, (ii) BASEVOCAB or SCIVOCAB. The two models using BASEVOCAB are fine-tuned from the corresponding BERT-base models. The other two models which use SCIVOCAB are trained from scratch. The experiments show that fine-tuning SciBERT on the English version of our dataset yielded outstanding classification performance.

AraBERT [4] (for Arabic data): is a pretrained language model based on BERT, specifically for the Arabic language, in the pursuit of achieving the same success that BERT did for the English language. The performance of AraBERT is compared to multilingual BERT from Google and other state-of-the-art approaches. Their results showed that the newly developed AraBERT achieved state-of-the-art performance on most tested Arabic NLP tasks.

Asafaya [42] (for Arabic data): is a pretrained language model based on BERT for Arabic. The final version of the corpus contains some non-Arabic words inline, which we did not remove from sentences since that would affect some tasks like Named Entity Recognition (NER). The model is not restricted to Modern Standard Arabic; the sentences contain some dialectical Arabic too.

A brief comparison of the three aforementioned models is shown in Table 2.1.

Table 2.1: Comparison Between BERT-based Models.

Model	**SciBERT (English)**	**AraBERT (Arabic)**	**Asafaya(Arabic)**
Size of corpus	123MB (1.14M papers)	24GB of text	95GB of text
Number of tokens/sentences	3.1 billion tokens	70 million sentences	8.2 billion words
Type of data	Papers from semanticscholar.org (full text and not just abstract)	• Arabic Wikipedia dump • 1.5B words Arabic corpus • OSIAN corpus • Asafir news articles • OSCAR unshuffled and filtered (added for new dataset)	• Arabic version of OSCAR - filtered from Common Crawl • Recent dump of Arabic Wikipedia • Other Arabic resources

Chapter 3

Approach

Since there is no available corpus of Arabic ETDs to be directly used, one of this project's goals is to build a seed corpus by collecting key metadata of Arabic ETDs such as abstracts, titles, and keywords. This will encourage more research to take place in this area.

In this research, the raw ETDs' metadata collected underwent a preprocessing phase to remove noisy and unwanted data. A classification model was then built and trained on 70% of the data before testing on the remaining 30%. Figure 3.1 shows the workflow of the Otrouha project.

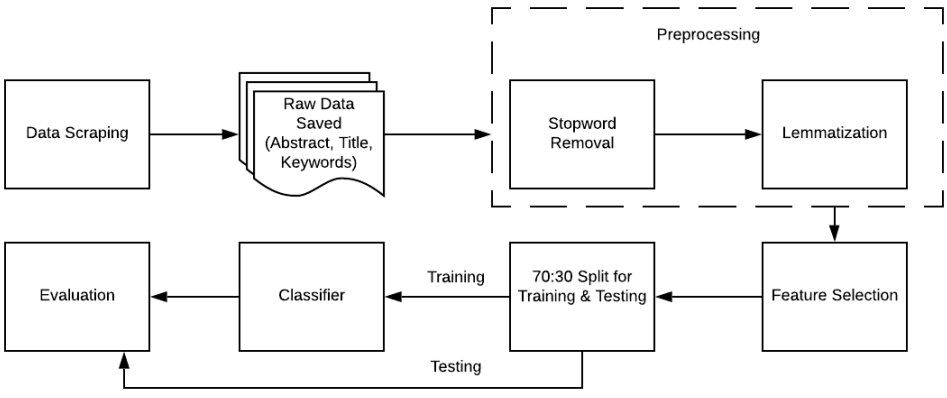

Figure 3.1: The workflow of the Otrouha methodology.

3.1 Dataset

3.1.1 Data Collection

After a thorough exploration, the Saudi Digital Library (SDL) was found to contain several databases such as ProQuest, AskZad, Saudi Cultural Mission in Australia, and Dar Almandumah. The AskZad component has the following advantages:

- It is rich in Arabic ETDs. According to AskZad [5] , it is considered to be the premier place for Arabic academic research. It contains direct image scans of original articles going as far back as 1823.
- In addition to scanned ETDs, it provides key metadata and associated bibliographies embedded in their website, which facilitates the data collection for this project.
- Its categorization system is compatible with categorization systems of other digital libraries, such as ProQuest.
- The bibliographies are available in different languages such as Arabic, English, and French.
- It is getting larger, i.e., the number of documents it includes is increasing, introducing more categories than the currently existing ones.

In order to collect data from the AskZad digital library, a web scrapper was designed and developed to perform the data collection process efficiently. The preliminary work led to collecting abstracts of 518 ETDs. Then, further improvements were made to the scrapper, which led to increasing the size of the corpus to contain metadata of more than 7000 ETDs. This metadata included not only the abstracts but also the titles and keywords. In addition,

the English version was also collected to aid in comparing the language impact on the classification performance.

3.1.2 Categories and Data Exploration

After analyzing the AskZad digital library to gain a deeper understanding of their categorization system, we noted it had 16 categories, where the number of ETDs in each category varies. For example, the *Education* category has about 8,700 ETDs, whereas the *Culture* category has about 190 ETDs. Each of the ETDs is assigned to one of the 16 categories. The preliminary collection covers a subset of that (i.e., 12 categories) since the number of ETDs in each of the other categories is small. The improved scrapper yielded collected data that included records from each of the 16 categories. The AskZad digital library categorization system keeps getting larger, which opens the possibility for more extensive research to take place in the future. The distribution of the number of records in each category is shown in Figure 3.2.

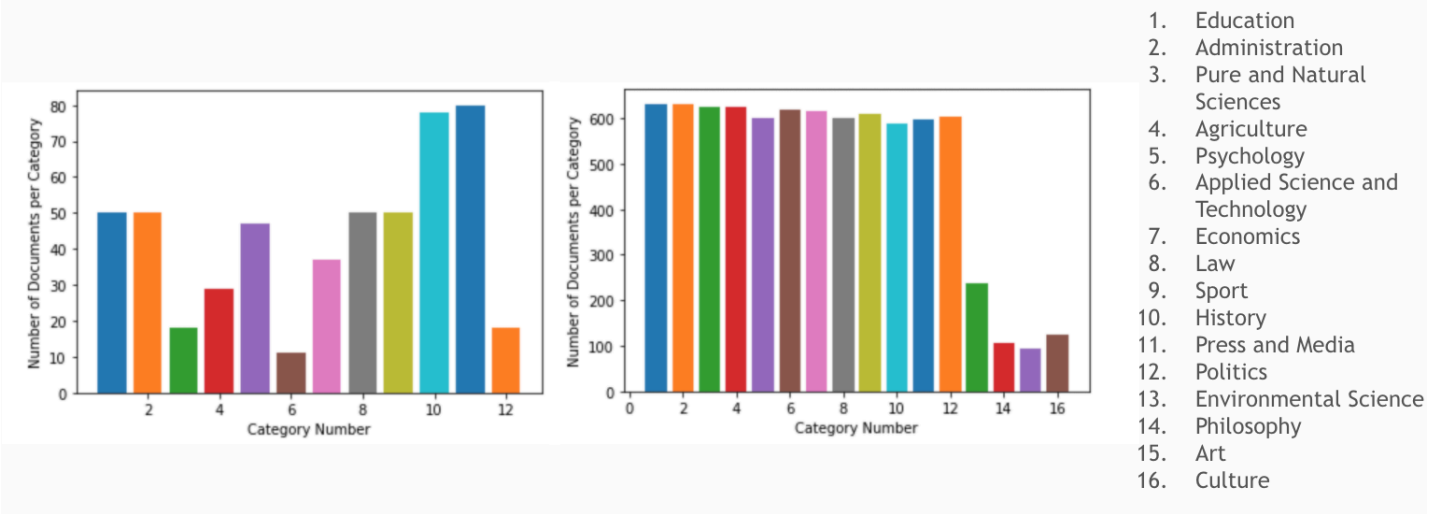

Figure 3.2: The Distribution of Records in Each Category in the Initially Collected Dataset and the Enlarged Dataset.

3.1.3 Mapping of AskZad Categories to ProQuest Categories

As mentioned earlier, one of the features that makes the AskZad digital library a suitable source of data for this project is its taxonomy. This taxonomy is simple and compatible with the taxonomy of ProQuest, which has the world's most comprehensive curated collection of multi-disciplinary dissertations and theses from around the world, offering over 5 million citations and 2.7 million full-text works from thousands of universities [37]. A detailed overview of the ProQuest categories can be found at [38].

Moreover, the AskZad digital library provides structured and consistent metadata for each ETD. This metadata is embedded in their website for ETDs that are scanned as well as ETDs that are born digital, which facilitates the task of collecting metadata directly without the need of using OCR (Optical Character Recognition).

In addition, the AskZad website is available in Arabic, English, and French versions. Therefore, its English version was used to get the translation of each category to manually map AskZad categories to ProQuest categories for 2018-2019, as shown in Figure 3.3.

For example, the *Law* category in AskZad is mapped to the high-level category *Law and Legal Studies* and its sub-categories in ProQuest. All categories in AskZad have been mapped to one or more corresponding categories in ProQuest, which ensures future consistency in categorization among different digital libraries of Arabic ETDs.

3.1.4 Data Pre-processing

In order to clean the raw data, some preprocessing techniques were applied. This included stopword removal and lemmatization. Stopwords were removed using a library from the

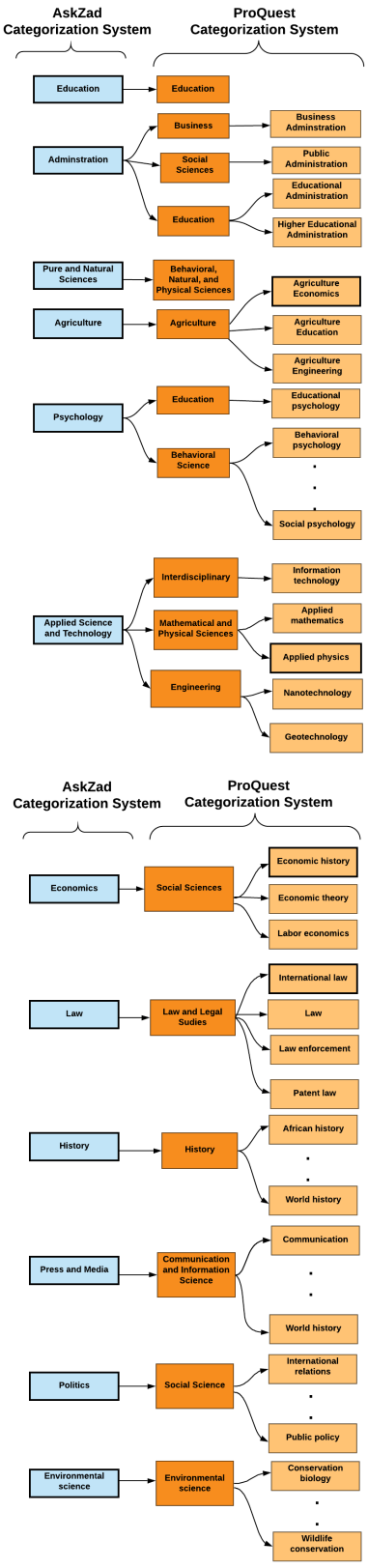

Figure 3.3: Mapping AskZad categories to ProQuest categories.

Natural Language Tool Kit (NLTK) The average number of words in a raw abstract originally was 204 words; after stopword removal, it was 168 words. Thus, approximately 17% of the raw data was removed for dimensionality reduction and better classification.

Different lemmatizers have been tested on a sample abstract to compare their performance. It was found that the Farasa lemmatizer [2] gave more accurate output and outperformed the state-of-the-art MADAMIRA [36] and Stanford Arabic segmenter in regard to Arabic segmentation and lemmatization. Farasa is a full-stack package to deal with Arabic Language Processing. It has a REST API that gets an HTTP POST request of the raw text and responds by returning the lemmatized text. To ensure the POST request returns valid data, an **A**dvanced **R**EST **C**lient (ARC) tool[1] has been used for manual testing. Figure 3.4 shows the online demo page of different modules for text processing, and Figure 3.5 shows a code snippet of using its lemmatization module.

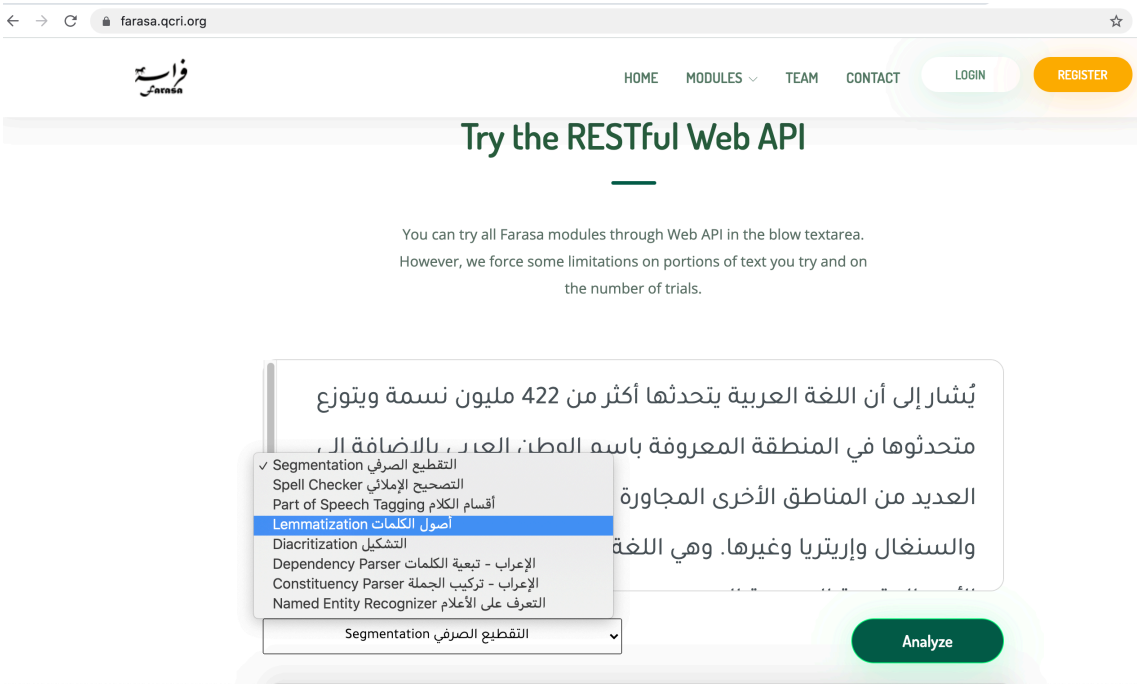

Figure 3.4: Farasa Online Demo.

[1] https://install.advancedrestclient.com/install

3.2 Classification Model

In order to provide automatic text classification that can improve the browsability and accessibility of Arabic ETDs, both classical supervised machine learning as well as deep learning classification techniques were tried. The collected data was split into training and testing in the ratios of 70% and 30%, respectively.

3.2.1 Classical Supervised Machine Learning

According to previous work done in the area of text classification, the following algorithms are commonly used: **S**upport **V**ector **M**achines (SVMs), **D**ecision **T**rees (DT), and **R**andom **F**orest (RF). Their performance depends on several parameters, such as the classification task (multiclass vs. binary) as well as the type, size, and language of the dataset.

The feature set of the data was extracted using the TF-IDF Vectorizer in the SciKit-learn

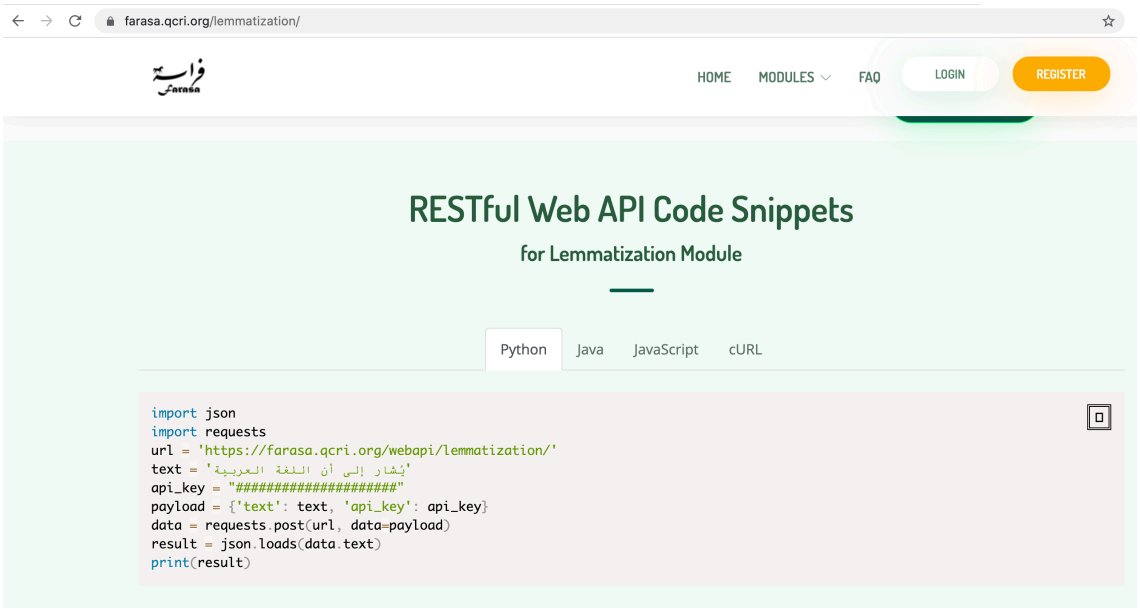

Figure 3.5: Farasa RESTful Web API Code Snippet for Lemmatization Module.

library. TF-IDF stands for Term Frequency-Inverse Document Frequency. The TF-IDF Vectorizer converts a text collection to a matrix with TF-IDF features. Since research in this area is scarce, both multiclass and binary classifications were conducted to compare the performance of these techniques for the Arabic ETDs classification task. The results are reported in Section 4.2.

Multiclass Classification: Three classifiers were tried, which are SVM, Decision Trees, and Random Forest. As shown in Chapter 4, the performance of each of the aforementioned classifiers on the Arabic data was poor, with an overall average accuracy of 24%. The initial insight from this experiment was that the dataset size contributed to the poor performance. Therefore, more data was collected, and the experiments were rerun.

Binary Classification (one-vs-rest): Binary classification, with the Random Forest classifier, was used to get more information about classification performance per category. Using binary classification resulted in better performance with a higher average accuracy per category of 68%. In order to determine how the dataset size would impact the classification performance, binary classification took place on the larger dataset collected. Different sizes of the dataset were trained and tested against the classifier. The results of these experiments are shown in Chapter 4.

3.2.2 Supervised Deep Learning

Since deep learning has been recently regaining attention in the text classification literature and giving promising results, different models were tried in order to provide a classification for the dataset of Arabic ETDs. Since this is exploratory research, different existing models were used as baselines to see how each of them performed with our dataset. These models

include **C**onvolutional **N**eural **N**etworks (CNN) and Pretrained Language Models.

CNN with Embedding Layer: This model worked well on the dataset of IMDB[2] which is the world's most popular and authoritative source for movie, TV, and celebrity content [33]. This model pads the sequences before feeding the neural network. CNN modeling starts off with an embedding layer, which maps vocabulary indices into embedding dimensions. Then it adds a Convolution1D, which will learn filters, followed by a max-pooling layer. After that, it projects onto a single unit output layer and compresses it with a sigmoid function. Then, the model is run using a binary cross-entropy loss function and Adam optimizer. To prevent overfitting of the CNN, a dropout regularization technique was adopted with different rates that range from 0.2 to 0.9. Dropout was used in three positions: after the embedding layer, pooling layer, and **F**ully-**C**onnected (FC) layer. Also, the Adam optimizer was used to train CNN. ReLU and sigmoid functions are used as activation functions for both convolution and output layers. This model resulted in an accuracy of 10% for our dataset, while for the IMDB dataset, it resulted in an accuracy of 89% after two epochs. Therefore, the following model was tried.

CNN without Embedding Layer: This CNN model was designed for intent classification for Arabic text data from a Twitter dataset [6]. It is different in the conversion of data from the aforementioned model, where it does not use an embedding layer, but the CNN architecture was the same. Using this model on our dataset resulted in an accuracy of approximately 12%.

Modified Model: Since the accuracy achieved using existing models was a maximum of 12%, several experiments, and parameter tuning, were tried in order to determine what

[2]https://developer.imdb.com/

could be negatively affecting the performance. To convert the text into numeric values, a "texts_to_sequences" Keras function was used. It transforms each text in the dataset into a sequence of integers, which can contribute to poor performance. For the neural network architecture, we removed the convolutional layer and used only the fully-connected neural network. This resulted in achieving a great improvement in the accuracy, to reach 87%. However, the F1, precision, and recall were all approximately 15%, which is very low. The potential reasons behind this are discussed in Chapter 4.

Pre-trained Language Models To benefit from deep learning, which offloads the challenge of feature engineering, and overcome the need for large-scale datasets to train the model, pretrained language models were used. In this research, BERT-based models were used that were pretrained on a large dataset, as described in Section 2.2.2, then adapted on the ETDs' dataset. SciBERT was adapted on the English version of the dataset, while AraBERT and Asafaya were adapted on the Arabic version of the dataset. There are two main paradigms for adaptation; feature extraction and fine-tuning. In feature extraction, the model's weights are frozen, and the pretrained representations are used in a downstream model similar to classic feature-based approaches [28]. Alternatively, a pretrained model's parameters can be unfrozen and fine-tuned on a new task [12]. The first paradigm was adopted in this research. This approach improved the classification performance significantly, as shown in Chapter 4.

3.3 Evaluation

To measure the performance of the classification model, classification accuracy has been used. However, it was found that accuracy is not always enough to correctly judge the

model, especially when the data is imbalanced. Accordingly, other performance metrics were calculated, which are precision, recall, and F1-score, in addition to the accuracy. Precision is the number of correct positive results divided by the number of positive results predicted by the classifier. The recall is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). F1-score tries to find the balance between precision and recall.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TotalSample}$$

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$F1 - score = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Chapter 4

Experimental Setup, Results, and Discussion

The first section of this chapter includes a detailed overview of the setup of the experiments and the performance metrics used. The second section presents the results of the experiments and a discussion.

4.1 Experimental Setup

The primary goal of our experimentation is to determine whether machine learning and deep learning techniques can be used for automatic classification of Arabic ETDs, using only their key metadata, and yield satisfactory results. The experiments are noted as 1 and 2 in the right column of Figure 4.1. This goal led to the need to understand each category. Therefore binary classification was explored after multiclass classification. In experiment 1, these two are noted as A and B. There are different variables that needed to be explored for multiclass classification and binary classification. Thus, the following sub-goals arose. The first is measuring how the difference in dataset size can affect classification performance for each of the classification approaches. Thus, in experiments 1.A and 1.B, the two dataset sizes are noted as I and II. The second sub-goal is determining whether using abstracts only versus abstract, title, and keywords together can yield better classification performance. This

was studied using the larger dataset, as part of experiment 1, for both 1.A and 1.B, noted further as II and III. The third sub-goal is whether vectorizing text using different values of maximum features would make a difference and whether this value changes as the dataset size changes. This was studied in experiment 1.B, in parts noted as IV and V, respectively.

4.2 Results and Discussion

4.2.1 Supervised Machine Learning

Multiclass Classification: The accuracies presented in Table 4.1 show the preliminary results of different classifiers, using the dataset that consisted of only 518 Arabic abstracts. The baseline performance for multiclass classification is 1/number of classes, which is random guessing. In this case, the random guessing would be 1/12, which is approximately 0.08. Both SVM and Decision Tree classifiers gave 0.24, while the Random Forest classifier gave 0.25. The approach of collecting data was improved to collect more data. As mentioned earlier, increasing the dataset size involved two approaches. One is vertical, by training using metadata of 7632 ETDs instead of 518. The other is horizontal, by adding the title and keywords corresponding to each abstract.

Considering the horizontal approach, the experiments were rerun once using abstracts only, and another time using abstracts, titles, and keywords together, to learn the impact of dataset size on the classification performance and whether titles and keywords would make a difference.

Considering the vertical approach, the larger the dataset, the more categories it covered. This led to classifying across 16 categories instead of 12. In this case, the random guessing

- Experiments
  - **Machine Learning** **1**
    - **Multiclass Classification** ..... A
      - Dataset 1: 518 Abstracts (12 Categories) I
      - Dataset 2: 7632 Abstracts (16 Categories) II
      - Dataset 2: 7632 Abstracts, Titles, and Keywords (16 Categories) III
    - **Binary Classification** ..... B
      - Using Default Max Features
        - Dataset 1: 518 Abstracts (12 Categories) I
        - Dataset 2: 7632 Abstracts (16 Categories) II
        - Dataset 2: 7632 Abstracts, Titles, and Keywords (16 Categories) III
      - Tuning Value of Max Features for Dataset of Size 7632 Abstracts, Titles, and Keywords) IV
      - Tuning Value of Max Features for Different Dataset Sizes of Abstracts, Titles, and Keywords) V
  - **Deep Learning** **2**
    - Dataset 1: 518 Abstracts (12 Categories) - IMDB Model ..... A
    - Dataset 2: 7632 Abstracts, Titles, and Keywords (16 Categories) - Pretrained Language Models ..... B

Figure 4.1: Experimental Setup.

would be 1/16, which is approximately 0.06.

Increasing the dataset size improved the accuracy overall to an average of 0.45 to 0.50 for both

SVM and Random Forest classifiers. Decision trees classifier performance degraded. Adding titles and keywords did not improve the classification performance but rather increased the dimensionality and negatively affected the performance. Tables 4.2 and 4.3 show the effect of enlarging the dataset size on the classification performance.

Table 4.1: Results of Experiment 1.A.I using Multiclass Classification with Dataset Size 518 (Abstracts Only, Across 12 Categories).

Classifier	Accuracy
Support Vector Machines (SVM)	0.24
Decision Trees	0.24
Random Forest (RF)	0.25

Table 4.2: Results of Experiment 1.A.II using Multiclass Classification with Dataset Size 7632 (Abstracts Only, Across 16 Categories).

Classifier	Accuracy
Support Vector Machines (SVM)	0.50
Decision Trees	0.20
Random Forest (RF)	0.51

Table 4.3: Results of Experiment 1.A.III using Multiclass Classification with Dataset Size 7632 (Abstracts, Titles, and Keywords, Across 16 Categories).

Classifier	Accuracy
Support Vector Machines (SVM)	0.45
Decision Trees	0.19
Random Forest (RF)	0.41

In order to understand more about the classification per category, experiments using binary classification took place.

Binary Classification: Among the classifiers tried in multiclass classification, the Random Forest classifier was found to give the highest accuracy. For that reason, it was chosen to conduct the second set of experiments, which was the binary classification (one-vs-all). More

evaluation metrics were included, such as precision, recall, and F1. Table 4.4 shows the result using the preliminary dataset collected. The baseline performance for binary classification is 0.50, which is random guessing. Binary classification resulted in much better performance than multiclass classification. In the preliminary results, among the 12 categories, eight of them achieved an accuracy of 0.50 or higher. Two of these eight categories yielded the highest accuracy as well as highest F-measure. These two categories are *Applied Science and Technology*, and *Economics*, where the highest accuracies were 0.95 and 0.80, respectively. The other six categories had satisfactory results for precision, recall, and F1. The remaining four categories resulted in less than 0.50 accuracy. The results of using the larger dataset are presented in Tables 4.5 and 4.6. Since some categories yielded unexpected results, one of the insights was that the max_features of TF-IDF Vectorizer could be affecting the results, especially since the number of samples is not equal across all the categories. Therefore, a new sub-goal arose, which is determining how the max_features parameter contributes to the performance. Various values were tried for the max_features parameter of the TF-IDF Vectorizer. The results are split across Tables 4.7 and 4.9 with the best max_features value appearing in bold.

Different dataset sizes were used to determine what choice of max_features yields the highest accuracy. The results are split across Tables 4.11 and 4.13.

Table 4.4: Results of Experiment 1.B.I using Binary Classification (One-vs-all) for Dataset Size 518 (Abstracts Only).

Category	Precision	Recall	F1	Accuracy	No. Docs
Administration	0.18	0.45	0.26	0.36	50
Agriculture	0.69	0.65	0.64	0.66	29
Applied Science & Technology	0.95	0.95	0.95	0.95	11
Economics	0.80	0.79	0.79	0.80	37
Education	0.69	0.66	0.65	0.66	50
Environmental Science	0.79	0.60	0.54	0.63	18
History	0.43	0.43	0.39	0.49	50
Law	0.57	0.57	0.53	0.53	50
Politics	0.30	0.37	0.37	0.60	80
Press & Media	0.42	0.42	0.42	0.46	78
Psychology	0.21	0.50	0.30	0.43	47
Pure & Natural Science	0.62	0.63	0.62	0.63	18

Table 4.5: Results of Experiment 1.B.II using Binary Classification (One-vs-all) for Dataset Size: 7632 (Abstracts Only).

Category	Precision	Recall	F1	Accuracy	No. Docs
Administration	0.66	0.66	0.66	0.66	605
Agriculture	0.84	0.85	0.84	0.84	599
Applied Science & Technology	0.68	0.68	0.68	0.68	592
Art	0.72	0.74	0.72	0.71	95
Culture	0.83	0.83	0.83	0.83	126
Economics	0.77	0.78	0.78	0.77	588
Education	0.83	0.68	0.82	0.82	605
Environmental Science	0.62	0.64	0.63	0.62	236
History	0.68	0.68	0.68	0.68	570
Law	0.76	0.76	0.75	0.75	573
Philosophy	0.51	0.52	0.52	0.51	105
Politics	0.61	0.62	0.62	0.61	585
Press & Media	0.60	0.60	0.60	0.60	581
Psychology	0.69	0.69	0.69	0.69	583
Pure & Natural Science	0.63	0.64	0.64	0.64	598
Sport	0.80	0.80	0.80	0.80	591

Table 4.6: Results of Experiment 1.B.III using Binary Classification (One-vs-all) for Dataset Size: 7632 (Abstracts, Titles, and Keywords).

Category	Precision	Recall	F1	Accuracy	No. Docs
Administration	0.83	0.83	0.83	0.83	605
Agriculture	0.70	0.72	0.69	0.69	599
Applied Science & Technology	0.52	0.52	0.52	0.52	592
Art	0.65	0.65	0.65	0.65	95
Culture	0.80	0.80	0.80	0.80	126
Economics	0.65	0.65	0.65	0.65	588
Education	0.85	0.85	0.85	0.85	605
Environmental Science	0.74	0.75	0.75	0.74	236
History	0.53	0.53	0.53	0.53	570
Law	0.86	0.87	0.86	0.86	573
Philosophy	0.62	0.62	0.62	0.62	105
Politics	0.77	0.79	0.77	0.76	585
Press & Media	0.70	0.73	0.70	0.68	581
Psychology	0.58	0.58	0.68	0.68	583
Pure & Natural Science	0.52	0.51	0.51	0.51	598
Sport	0.88	0.88	0.88	0.88	591

Table 4.7: Results of Experiment 1.B.IV for Tuning Max Features Parameter of TF-IDF Vectorizer of Binary Classification on Dataset Size 7632 for the First 8 Categories.

Category	Max Features	Precision	Recall	F1	Accuracy	Samples
Administration	30	0.73	0.73	0.73	0.73	605
	50	**0.85**	**0.85**	**0.85**	**0.85**	
	100	0.98	0.77	0.77	0.77	
	200	0.82	0.82	0.82	0.82	
Agriculture	30	0.82	0.81	0.81	0.81	599
	50	**0.92**	**0.91**	**0.91**	**0.91**	
	100	0.69	0.65	0.63	0.65	
	200	0.63	0.72	0.63	0.59	
Applied Science & Technology	**30**	**0.81**	**0.81**	**0.81**	**0.81**	592
	50	0.75	0.75	0.75	0.75	
	100	0.46	0.46	0.46	0.46	
	200	0.62	0.62	0.62	0.62	
Art	30	0.68	0.68	0.68	0.68	95
	50	**0.72**	**0.72**	**0.72**	**0.72**	
	100	0.68	0.69	0.68	0.68	
	200	0.67	0.66	0.67	0.67	
Culture	30	0.77	0.77	0.77	0.77	126
	50	0.82	0.81	0.81	0.81	
	100	**0.85**	**0.84**	**0.84**	**0.84**	
	200	0.80	0.80	0.80	0.80	
Economics	30	0.54	0.54	0.54	0.54	588
	50	0.70	0.70	0.70	0.70	
	100	**0.75**	**0.75**	**0.75**	**0.75**	
	200	0.61	0.59	0.58	0.59	
Education	**30**	**0.90**	**0.90**	**0.90**	**0.90**	605
	50	0.63	0.61	0.60	0.62	
	100	0.75	0.75	0.75	0.75	
	200	0.83	0.83	0.82	0.82	
Environmental Science	30	0.46	0.46	0.45	0.45	236
	50	0.72	0.69	0.67	0.67	
	100	**0.77**	**0.75**	**0.73**	**0.73**	
	200	0.37	0.41	0.35	0.38	
History	**30**	**0.83**	**0.82**	**0.82**	**0.82**	570
	50	0.64	0.64	0.64	0.64	
	100	0.81	0.80	0.80	0.80	
	200	0.62	0.60	0.58	0.61	

Table 4.9: Continued Results of Experiment 1.B.IV for Tuning Max Features Parameter of TF-IDF Vectorizer of Binary Classification on Dataset Size 7632 for the Remaining 8 Categories.

Category	Max Features	Precision	Recall	F1	Accuracy	Samples
Law	30	0.54	0.54	0.54	0.54	573
	50	**0.95**	**0.95**	**0.95**	**0.95**	
	100	0.69	0.69	0.69	0.69	
	200	0.79	0.77	0.76	0.76	
Philosophy	30	0.39	0.39	0.38	0.38	105
	50	**0.66**	**0.66**	**0.66**	**0.66**	
	100	0.45	0.45	0.44	0.44	
	200	0.51	0.51	0.45	0.48	
Politics	30	0.80	0.76	0.76	0.77	585
	50	0.83	0.82	0.82	0.82	
	100	**0.91**	**0.91**	**0.91**	**0.91**	
	200	0.88	0.86	0.87	0.87	
Press & Media	30	0.68	0.68	0.68	0.67	581
	50	0.70	0.79	0.70	0.70	
	100	**0.82**	**0.82**	**0.82**	**0.82**	
	200	0.72	0.71	0.71	0.71	
Psychology	30	0.76	0.75	0.75	0.75	583
	50	**0.79**	**0.79**	**0.79**	**0.79**	
	100	0.72	0.72	0.72	0.72	
	200	0.63	0.63	0.63	0.63	
Pure & Natural Sciences	**30**	**0.67**	**0.66**	**0.65**	**0.66**	598
	50	0.54	0.54	0.54	0.54	
	100	0.53	0.53	0.54	0.54	
	200	0.60	0.58	0.57	0.59	
Sport	30	0.84	0.84	0.84	0.84	591
	50	0.69	0.69	0.69	0.69	
	100	**0.93**	**0.93**	**0.93**	**0.93**	
	200	0.79	0.79	0.70	0.79	

Table 4.11: Results of Experiment 1.B.V for Best Max Features Values of TF-IDF Vectorizer of Binary Classification for Different Sizes of Data for the First 8 Categories.

Category	Dataset Size	Best Max_Features
Administration	100%	50
	75%	50
	50%	50
	25%	30
Agriculture	100%	50
	75%	100
	50%	100
	25%	50
Applied Science and Technology	100%	30
	75%	30
	50%	100
	25%	200
Art	100%	50
	75%	30
	50%	200
	25%	30
Culture	100%	100
	75%	30
	50%	100
	25%	30
Economics	100%	100
	75%	30
	50%	30
	25%	50
Education	100%	30
	75%	100
	50%	50
	25%	50
Environmental Science	100%	100
	75%	30
	50%	30
	25%	200
History	100%	30
	75%	50
	50%	50
	25%	30

4.2.2 Deep Learning

Supervised Deep Learning

The average results of deep learning for Experiment 2.A (compared with the application to a different dataset) using the fully-connected layers model are shown in Table 4.15. This model with Keras includes three dense layers using a **R**ectified **L**inear **U**nits (ReLU) activation function followed by a dropout layer and a final dense layer using softmax activation. Using this architecture, an accuracy ranging between 0.85-0.88 was achieved. However, the F1, precision, and recall percentages remained very low, ranging between 0.10-0.20. Some additional research has been done on the interpretation of these values. It was conjectured that the reason for having high accuracy and a low F1 is the imbalance in the dataset, which leads to skewness and degrades the model's performance [29]. To further test this argument, the dataset distribution was studied. It was found that the *Politics* class was the most commonly predicted one, which contains the highest number of metadata records. That is the reason for the discrepancy between accuracy and F1. In order to deal with the imbalanced data problem, more data was collected. Aso, another method was used, transfer learning using pretrained language models, as shown in the following subsection.

Pretrained Language Models (BERT-based)

The SciBERT pretrained language model was tried using the English version of the data, then the AraBERT and Asafya models were tried on the Arabic version. Table 4.16 shows the precision, recall, and F1 scores for each model. Also, in order to have a closer look at the performance, confusion matrices were generated for each of the models, as shown in Figures 4.2, 4.3, and 4.4, respectively. A confusion matrix allows visualization of the performance of

Table 4.13: Continued Results for Experiment 1.B.V for Best Max Features Values of TF-IDF Vectorizer of Binary Classification for Different Sizes of Data for the Remaining 8 Categories.

Category	Dataset Size	Best Max_Features
Law	100%	50
	75%	30
	50%	30
	25%	100
Philosophy	100%	50
	75%	50
	50%	30
	25%	100
Politics	100%	50
	75%	200
	50%	30
	25%	30
Press and Media	100%	100
	75%	100
	50%	30
	25%	50
Psychology	100%	50
	75%	100
	50%	50
	25%	50
Pure & Natural Sciences	100%	30
	75%	100
	50%	50
	25%	100
Sport	100%	100
	75%	100
	50%	30
	25%	30

Table 4.15: Results for Experiment 2.A for Using a Deep Learning Model for Dataset Size 518 (Abstracts Only).

Model	Precision	Recall	F1	Accuracy
Reference model (IMDB)	0.96	0.92	0.94	0.97
Modified model	0.15	0.15	0.15	0.87

an algorithm and illustrates if a certain class is usually misclassified as another class; this can help guide future fine-tuning. For example, using SciBERT, the *Culture* class has 42 records in the test set; 20 of them were correctly classified, while 12 of them were misclassified as *Press and Media*. This can give an idea of which classes' features are relatively close to others.

Table 4.16: Results of Experiment 2.B for Using BERT-based Language Models on English and Arabic Versions of Data of Size 7632 (Abstracts, Titles, and Keywords).

Model	Precision	Recall	F1	Accuracy
SciBERT (English Data)	0.92	0.92	0.92	0.91
AraBERT (Arabic Data)	0.84	0.83	0.83	0.84
Asafaya (Arabic Data)	0.83	0.83	0.82	0.83

```
[[168   9   0   0   6   0   0   0   2   0   1   1   0   0   0   0]
 [  4 187   0   0   1   2   6   0   4   0   0   1   0   0   0   0]
 [  0   0 153   4   0  14   1   0   0   0   0   0   9   0   0   0]
 [  0   1  12 173   0   0   3   0   0   0   0   0   0   0   0   0]
 [ 12   0   1   0 159   0   0   0   0   0   0   2   0   0   0   0]
 [  0   1   4   0   0 162   1   0   0   1   0   1   2   0   0   0]
 [  0   4   1   1   0   2 179   1   0   0   1   1   1   0   0   0]
 [  0   0   0   0   1   0   0 186   1   0   0   1   1   0   0   0]
 [  2   6   0   0   1   1   0   0 163   0   0   0   0   0   0   0]
 [  0   0   0   0   0   1   0   0   0 172   0   4   0   2   0   5]
 [  0   1   0   0   0   0   1   0   0   1 170   0   0   0   0   0]
 [  0   0   0   0   0   0   0   2   0   0   3 176   0   0   0   0]
 [  0   0   4   2   0   1   1   1   0   0   1   0  74   0   0   0]
 [  0   0   0   0   1   0   0   1   0   0   1   0   0  27   0   0]
 [  0   0   1   0   0   0   0   0   0   3   3   0   0   0  16   0]
 [  0   0   1   0   1   0   0   0   0   3  12   0   0   0   0  20]]
```

Figure 4.2: Confusion Matrix When Using SciBERT on Data of Size 7632 (Abstracts, Titles, and Keywords).

The observations suggest that deep learning using generic pretrained language models outper-

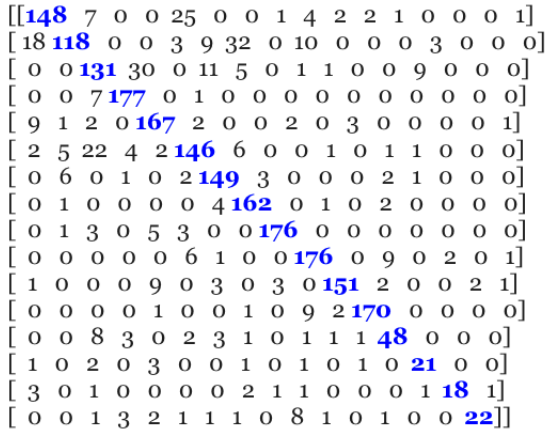

Figure 4.3: Confusion Matrix When Using AraBERT on Data of Size 7632 (Abstracts, Titles, and Keywords).

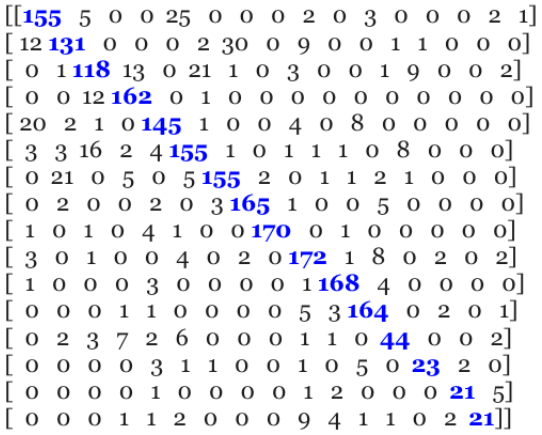

Figure 4.4: Confusion Matrix When Using Asafaya on Data of Size 7632 (Abstracts, Titles, and Keywords).

forms classical machine learning techniques and deep learning without pretrained language models. The data imbalance persisted even after enlarging the dataset, where the number of records in four categories was much smaller than the other twelve categories. However, using pretrained language models aided in overcoming this problem for the underrepresented categories, such as *Philosophy*, *Art*, *Culture*, and *Sport*. This is due to using the representations generated by the previously trained network to extract meanings from the ETD dataset.

For example, *Philosophy* has 68 records in the test set, where 48 of them were classified correctly. Using pretrained language models leads to overcoming the lack of large-scale data needed for deep learning training and reduces the time needed for training from scratch. BERT-based language models are designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both the left and right context in all layers. Since the Arabic language is written from right to left, conditioning on both left and right context in all layers improved the classification results. In addition, the size of the dataset is believed to be contributing to the robustness of the results since the deep learning algorithm's performance increases as the data size increases. Moreover, the results show that using SciBERT on the English version of data yielded higher performance than using AraBERT and Asafaya on the Arabic data, although it is trained on a much smaller data size. This is highly likely due to using domain-adaptive pretraining of language models, where SciBERT is pretrained on scientific data rather than generic data like AraBERT and Asafaya.

Chapter 5

Conclusions

As the number of Arabic Electronic Theses and Dissertations increases, the need increases to make them more browsable and accessible. One of the ways to achieve this is by providing automatic subject classification. Since there is limited available open access to Arabic ETDs, this research aims to build a corpus of metadata of Arabic ETDs and to research different approaches to automatically classify them using various machine learning and deep learning techniques. Metadata of approximately 7600 ETDs was collected from the AskZad digital library. The initial experiments were done using only abstracts of ETDs. Then more metadata was collected for more ETDs. The hypotheses upon which this research was built are discussed below. The first is that machine learning and natural language processing techniques can automatically classify Arabic ETDs, based on their metadata. This hypothesis is proven true, where binary classification (one-vs-all) gave better results than multiclass classification. The second is that regarding selecting a suitable classification method, the hypothesis is that automatic classification using deep learning would not be suitable if the size of the corpus is not huge. This hypothesis is proven false. Generic deep learning architectures did not yield reliable results. However, using pretrained language models such as BERT-based models gave promising results despite the relatively small dataset size needed for deep learning. This suggests that understanding languages is essential for the classification of text.

The third hypothesis is that the more documents the corpus contains, the more accurate the automatic subject classification becomes. This hypothesis is proven true, where the results

show that increasing the dataset size was vital in improving the outcomes.

Fourth, regarding what data is needed, the hypothesize is that if more data is added to each dataset document, the accuracy of the automatic subject classification will increase. For example, classifying by subject category, using abstracts only might do less well than when abstracts, titles, and keywords are used together. The results show that this hypothesis is false, and adding more metadata did not significantly improve the classification performance.

Fifth, regarding how this works with Arabic vs. other languages, the hypothesis is that the language itself can contribute to the performance of the classification model. For example, automatically classifying the Arabic version of the metadata would result in a different performance than classifying the English version. This hypothesis is proven true since SciBERT, which was adapted on the English version of the dataset, gave higher performance than AraBERT and Asafaya, which were adapted on the Arabic version of the dataset.

The corpus built as part of this project is expected to encourage more research on Arabic ETDs, which is a rich genre of data.

Chapter 6

Future Work

This research, along with the experiments and results, provided good insights on how to proceed when moving forward.

Collecting more Arabic ETDs is essential for improving the machine learning process. This can include collecting metadata of ETDs and full-length ETDs. A challenge will be that not all ETDs are born digital, but rather were captured as a scanned version. This can open more research to extract the text using OCR (Optical Character Recognition) techniques and convert the text into a machine-readable form.

Also, multilabel classification can be considered, but expert annotation will be needed to fine-tune AskZad categories to map to ProQuest categories.

Pretrained language models, such as BERT, gave the highest classification performance on the automatic subject classification of Arabic ETDs relative to using classical machine learning techniques and to using generic deep learning architecture as discussed in Sections 4.2.1 and 4.2.2. Future work would include more fine-tuning to models pretrained on Arabic data to yield higher classification accuracy for Arabic ETDs.

Also, an Arabic equivalent to SciBERT can be developed to help enhance the browsability of academic work in Arabic. SciBERT has its vocabulary (SCIVOCAB), trained on cased and uncased versions, but this will not be needed for Arabic since it has no uppercase and lowercase versions of alphabets.

This study should encourage researchers planning to work on Arabic ETDs for further investigation and bench-marking in NLP.

Bibliography

[1] Mohammad AR Abdeen, Sami AlBouq, Ahmed Elmahalawy, and Sara Shehata. A Closer Look at Arabic Text Classification. *Int. J. Adv. Comput. Sci. Appl*, 10(11): 677–688, 2019.

[2] Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 11–16, 2016.

[3] S Al-Harbi, A Almuhareb, A Al-Thubaity, MS Khorsheed, and A Al-Rajeh. Automatic Arabic Text Classification. 9th International Conference on the Statistical Analysis of Textual Data (JADT 2008), 2008.

[4] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based Model for Arabic Language Understanding. *arXiv preprint arXiv:2003.00104*, 2020.

[5] AskZad. AskZad: The World's First and Largest Arabic Digital Library. http://askzad.com, 2020. [Accessed 14 June 2020].

[6] Abdallah Bashir. Intent Classification Module for Arabic Text Data Using CNN. https://github.com/abdallah197/text_classification_CNN_arabic, 2018. [Accessed 14 June 2020].

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv preprint arXiv:1903.10676*, 2019.

[8] Mohamed Boudchiche, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. Alkhalil Morpho Sys 2: A Robust Arabic Morpho-syntactic Analyzer. *Journal of King Saud University-Computer and Information Sciences*, 29(2):141–146, 2017.

[9] Samir Boukil, Mohamed Biniz, Fatiha El Adnani, Loubna Cherrat, and Abd Elmajid El Moutaouakkil. Arabic Text Classification Using Deep Learning Technics. *International Journal of Grid and Distributed Computing*, 11:103–114, 2018.

[10] Jason Brownlee. Machine learning mastery. https://machinelearningmastery.com/types-of-classification-in-machine-learning/, 2020. [Accessed 26 November 2021].

[11] Abdelghani Dahou, Mohamed E. Abd Elaziz, Junwei Zhou, and Shengwu Xiong. Arabic Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm. *Computational Intelligence and Neuroscience*, 2019, 2019.

[12] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28:3079–3087, 2015.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Rehab M Duwairi. A Distance-based Classifier for Arabic Text Categorization. In *DMIN*, pages 187–192, 2005.

[15] Rehab M. Duwairi. Arabic Text Categorization. *International Arab Journal of Information Technology*, 4(2):125–131, 2007.

[16] Alaa Mustafa El-Halees. Arabic Text Classification Using Maximum Entropy. *IUG Journal of Natural Studies*, 15:157–167, 2007.

[17] Tarek El-Shishtawy and Fatma El-Ghannam. A Lemma Based Evaluator for Semitic Language Text Summarization Systems. *arXiv preprint arXiv:1403.5596*, 2014.

[18] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: Small sample performance. Technical report, California Univ Berkeley, 1952.

[19] Tarek F. Gharib, Mena B. Habib, and Zaki T. Fayed. Arabic Text Classification Using Support Vector Machines. *Int. J. Comput. Their Appl.*, 16:192–199, 2009.

[20] M. Hagiwara. *Real-World Natural Language Processing: Practical Applications with Deep Learning*. Simon and Schuster, 2021. ISBN 9781617296420. URL https://books.google.com/books?id=Ok5NEAAAQBAJ.

[21] Faten Khalfallah Hammouda and Abdelsalam Almarimi. Heuristic Lemmatization for Arabic Texts Indexation and Classification. *Journal of Computer Science*, 6:660–665, 2010.

[22] Fouzi Harrag and Eyas Al-Qawasma. Improving Arabic Text Categorization Using Neural Network with SVD. *J. Digit. Inf. Manag.*, 8:233–239, 2010.

[23] Vasu Jindal. A Personalized Markov Clustering and Deep Learning Approach for Arabic Text Categorization. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 145–151, 2016.

[24] Anjali Ganesh Jivani et al. A Comparative Study of Stemming Algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[25] Subbu Kannan, Vairaprakash Gurusamy, S Vijayarani, J Ilamathi, M Nithya, S Kannan, and V Gurusamy. Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2014.

[26] Laila Khreisat. Arabic Text Classification Using N-Gram Frequency Statistics: A Comparative Study. *DMIN*, 2006:78–82, 2006.

[27] Divya Khyani, BS Siddhartha, NM Niveditha, and BM Divya. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology*, 2020.

[28] Philipp Koehn, Franz J Och, and Daniel Marcu. Statistical phrase-based translation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 2003.

[29] Will Koehrsen. Beyond Accuracy: Precision and Recall. https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c, 2018. [Accessed 14 June 2020].

[30] Rim Koulali, Mahmoud El-Haj, and Abdelouafi Meziane. Arabic Topic Detection Using Automatic Text Summarisation. *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–4, 2013.

[31] Mohamed El Kourdi, Amine Bensaid, and Tajje-Eddine Rachidi. Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 51–58, 2004.

[32] Ben Lutkevich. How Language Modeling Works. https://www.techtarget.com/

searchenterpriseai/definition/language-modeling, 2020. [Accessed 14 June 2021].

[33] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

[34] NLTK. NLTK: Natural Language Toolkit. https://www.nltk.org, 2009. [Accessed 26 November 2021].

[35] William S Noble. What Is a Support Vector Machine? *Nature biotechnology*, 24(12): 1565–1567, 2006.

[36] Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 26–31, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

[37] ProQuest. ProQuest Dissertations & Theses Global: The world's most comprehensive collection of multi-disciplinary dissertations and theses. https://about.proquest.com/en/products-services/pqdtglobal, 2021. [Accessed 26 November 2021].

[38] ProQuest. ProQuest Subject Categories. https://pq-static-content.proquest.

com/collateral/media2/documents/subject-categories-academic.pdf, 2021. [Accessed 26 November 2021].

[39] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[40] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer Learning in Natural Language Processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, 2019.

[41] S Rasoul Safavian and David Landgrebe. A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[42] Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. BERT-CNN for Offensive Speech Identification in Social Media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation, KUISAIL at SemEval-2020 Task 12*, pages 2054–2059, Barcelona (online), December 2020. International Committee for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.semeval-1.271.

[43] SakhrSoftware. Sakhr Software: Arabic language technology (Sakhr Solutions: Ranked Number 1 in Accuracy and Performance, Powered by the World's Leading Research in Arabic Natural Language Processing (NLP)). http://www.sakhr.com, 2022. [Accessed 6 January 2022].

[44] Hassan Sawaf, Jorg Zaplo, and Hermann Ney. Statistical Classification Methods for Arabic News Articles. In *Third Arabic Natural Language Processing Workshop, in ACL2001*, 2001.

[45] Catarina Silva and Bernardete Ribeiro. The Importance of Stop Word Removal on Recall Values in Text Categorization. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1661–1666. IEEE, 2003.

[46] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[47] Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. Naïve Bayes. *Encyclopedia of Machine Learning*, 15:713–714, 2010.