UNIVERSITY OF LEEDS

Compression versus Machine Learning for Classifying Modern Arabic Code-Switching in Social Media and Classical Arabic Hadith

Taghreed Tarmom

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds

Faculty of Engineering and Physical Sciences

School of Computing

October 2022

Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

Taghreed Tarmom

Abstract

This thesis aims to enrich Arabic resources by building several Arabic corpora and making them freely available to the Arabic research community. Therefore, the Bangor Arabic–English code-switching (BAEC) corpus, the Saudi Dialect Corpus (SDC) and the Egyptian Dialect Corpus (EDC) and the Non-Authentic Hadith (NAH) corpus were built.

This thesis carries out the detection of code-switching in Arabic varieties and dialects from social media platforms to evaluate the prediction by partial matching (PPM) compression approach, comparing it with a the support vector machine (SVM) classifier with character-based and word-based approaches. The aim was to test the PPM compression on modern standard Arabic (MSA) and Arabic dialect before using it on Hadith.To the best of our knowledge, no previous study involving the detection of code-switching between Arabic and English using PPM compression has been published before. The experimental results show that PPM compression achieved a higher accuracy rate than the SVM classifier when the training corpus correctly represented the language or dialect being studied.

Then, classifying experiments on Arabic Hadith to evaluate the PPM compression approach and compare it against machine learning and deep learning approaches was also performed. The aim was to classify Arabic Hadith into two main classification tasks: Hadith components classification and Hadith authenticity classification. For the former, the experimental results show that deep learning classifiers can achieve a higher classification accuracy than the other classifiers under study. However, the execution time for deep learning classifiers was high. For the latter, the experimental results showed that Isnad was the part of a Hadith resulting in the most effective automatic determination of authenticity. In addition, the results proved that Matan can be used to judge Hadiths with up to 85% accuracy. These experiments were novel in their approaches to Hadith authenticity classification because they investigated the use of the character-based text compression scheme PPM and DL classifiers.

Finally, the current thesis also investigated the automatic segmentation of Arabic Hadith using PPM compression. The experiments showed that PPM was effective in segmenting Hadith into its two main components, having been tested on different Hadith corpora that have different structures. The main innovation in these experiments was their use of a character-based text compression method to segment the Hadiths.

Publications associated with this research

The chapters in this thesis are based on several publications that have been published in the fields of Computer Science and Computational Linguistics.

- **Chapter 3**

  This chapter includes corpus discerptions from the following papers:

  1. *Tarmom T; Teahan W; Atwell E; Alsalka M (2019). Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The International Corpus Linguistics Conference 2019.*

  2. *Tarmom T; Teahan W; Atwell E; Alsalka M (2020). Compression versus traditional machine learning classifiers to detect code-switching in Varieties and Dialects: Arabic as a case study. Natural Language Engineering, pp. 1–14.*

  3. *Tarmom T; Atwell E; Alsalka MA (2020). Non-authentic Hadith Corpus: Design and Methodology. International Journal on Islamic Applications in Computer Science And Technology. 8(3), pp. 13–19*

  4. *Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.*

- **Chapter 4**

  This chapter is an extension of the following published papers:

  1. *Tarmom T; Teahan W; Atwell E; Alsalka M (2020). Compression versus Traditional*

Machine Learning Classifiers to Detect Code-Switching in Varieties and Dialects: Arabic as a Case Study. Natural Language Engineering, pp. 1–14.

2. *Tarmom T; Teahan W; Atwell E; Alsalka M (2019). Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The International Corpus Linguistics Conference 2019.*

- **Chapter 5**

  Part of this chapter is based on the following paper:

  1. *Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.*

- **Chapter 6**

  This chapter is based on the following paper:

  1. *Tarmom T; Atwell E; Alsalka MA (2020). Automatic Hadith Segmentation using PPM Compression. In Proceedings of the 17th International Conference on Natural Language Processing (ICON), pp. 22–29.*

  2. *Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.*

To

My journey companions and its fuel

The source of my inspiration

Leen,

Mohammad

and Salman

With Love

Acknowledgements

In the Name of Allah, the Most Gracious, the Most Merciful

All praise and thanks to Allah for guiding me toward the right path and helping me to complete this thesis. My thanks also go to many people for their encouragement, help and support during my studies.

Words fail to express my gratitude to my supervisors, Pro. Eric Atwell and Dr. Mohammad Ammar Alsalka, for their guidance, constant encouragement and help during the journey of preparing this thesis.

My deepest gratitude also goes out to my parents, who can never be thanked enough, for their love, endless support and prayers. At the same time, I would like to express my love and thanks to my children, 'the beats of my heart', Leen, Mohammad and Salman, who are the only source of inspiration to me.

My sincere gratitude and thank is also directed to my friend, Fatima, for her support and encouragement.

I would also like to acknowledge my country, the Kingdom of Saudi Arabia, which gave me the opportunity to study in the United Kingdom.

Contents

List of Figures

List of Tables

Abbreviations

BAEC Bangor Arabic–English Code-switching Corpus

SDC Saudi Dialect Corpus

EDC Egyptian Dialect Corpus

NAH Non-Authentic Hadith

MSA Modern Standard Arabic

CA Classical Arabic

NLP Natural Language Processing

PPM Prediction by Partial Matching

CML Classical Machine Learning

SVM Support Vector Machine

SMO Sequential Minimal Optimization

NB Naïve Bayes

DT Decision Tree

DL Deep Learning

LSTM Long-Short Term Memory

CNN Convolutional Neural Network

Chapter 1

Introduction

Although the primary religious text of Islam is the Quran, the Hadith is the second source of instruction. The Hadith includes references to any action, spoken order or implied approval of the holy Prophet Muhammad. Both of these sources of Islamic teaching are written in Classical Arabic (CA), which is entirely different from Modern Standard Arabic (MSA) in both its vocabulary and spelling. The individual Hadiths were delivered through a chain of narrators. Thus, each Hadith has an 'Isnad', or chain of narrators, and a 'Matan', or a description of an act of the Prophet Muhammad. In contrast to the Quran, some Hadiths, which have been handed down over the centuries, have been corrupted by narrators whose ability to transmit them accurately can be questioned. These Hadiths have been classified by Hadith scholars as non-authentic.

Text classification is the process of classifying a set of texts into one of a number of predefined classes. Text segmentation, on the other hand, is the process of identifying sentence boundaries in a given text and dividing them into their components. Several methods can be used for both tasks, such as prediction by partial matching (PPM) compression-based algorithm, classical machine learning (CML) algorithms and deep learning (DL) algorithms. Most Arabic natural language processing (NLP) studies concentrate on MSA, such as as the research by Khreisat (2009),Duwairi et al. (2009) and Alwedyan et al. (2011). However, there is a paucity of research on the classification and segmentation of CA texts, such as Hadith.

Hadiths are important in all aspects of Muslims' lives. Recently, there has been an increase in the spread of many forged Hadiths. This increases the importance of using NLP methods to determine the authenticity of Hadiths and classify them. A relatively small number of studies segment Arabic Hadith into its components and classify Arabic Hadiths in terms of both the topics covered and the Hadith's authenticity. As reported by Al -Kabi et al. (2014), a factor that contributes to this lack of research is that classifying Arabic Hadiths is more difficult than classifying other Arabic textual documents because each Hadith consists of two main elements: Isnad and Matan. In addition, Naji Al-Kabi et al. (2005) pointed out that the most challenging task of classifying Arabic Hadiths is that some may belong to more than one topic. Also, to the best of our knowledge, no study has concentrated on segmenting and classifying Hadiths using PPM compression method and DL methods. The current project seeks to address this gap in the research.

Another part of the current study addresses the automatic detection of code-switching in Arabic text. The aim was to test the PPM compression on MSA and Arabic dialect before using it on Hadith. Code-switching in written natural language text occurs when the author chooses to switch from one language to at least one other. This phenomenon is of particular interest when processing the Arabic language because of its frequent occurrence and because of the many dialects of Arabic in use. When processing Arabic text, it is important to identify where and when code-switching occurs, because more appropriate language resources can be applied to the task, thus making a significant improvement in processing performance. However, relatively few studies of code-switching for Arabic texts exist, not only regarding its frequency, but also regarding the development of software to automatically identify occurrences. A contributory factor is that dialect identification for Arabic has been found to be more difficult than for other languages. The current study also seeks to address this gap in the research.

1.1 Aims and Objectives

The overall aim of the present research is to study the performance of compression PPM against CML and DL algorithms when attempting to auto-classify Arabic text, specifically text from social media platforms and Hadith-related websites.

The specific objectives of the research are as follows:

1. Create a corpus of Arabic Hadith using Hadith websites. The corpus will be annotated to determine different features of the Hadiths, such as the Isnad, Matan, topics and

authenticity based on Hadith expert sources and to provide a ground truth. This will be achieved by paid annotators.

2. Create corpus of Arabic code-switching texts by using samples obtained from online sources such as Facebook and then annotate these corpora to determine the occurrence of code-switching to provide a ground truth.
3. Create new corpora containing samples of text from different Arabic dialects such as Egyptian and Saudi to train Egyptian and Saudi models.
4. Adapt the PPM compression-based approach in different applications (using the corpora that were created for previous objectives), such as
   (a) Detect code-switching in varieties and dialects
   (b) Hadith components categorisation
   (c) Hadith segmentation of Isnad and Matan
   (d) Hadith authenticity classification
5. Compare the compression-based approach, classical ML classifiers and the DL classifiers to the automatic classification of Arabic Hadith.
6. Compare the execution time of these methods to discover the faster and slower method of classifying Arabic Hadith.

The sub-objectives are as follows:

- Investigate the best order *o* of PPM for Arabic Hadith segmentation, where *o* is the number of characters used for predication.
- Conduct a performance evaluation of the CML for text classification using different features such as word and character features.
- Identify the part of a Hadith (Isnad, Matan or both) that works the best for effective automatic determination of authenticity.
- Study the impact of training parameters such as training corpus size on the classification results.

1.2 Research Question

The specific research questions underlying this research are as follows:

- *How does the effectiveness of the PPM compression-based approach compare against classical ML and DL algorithms for classifying Arabic Hadith text according to Hadith components and authenticity?*
- *How does the effectiveness of the PPM compression-based approach compare against traditional CML algorithms for detecting code-switching in varieties and dialects from social media platforms?*

1.3 Contributions

The main contributions that have been achieved in the fields of Arabic text classification and NLP are as follows:

1. The foremost contribution is the construction of several Arabic corpora and make them freely available to the Arabic research community. First, an Arabic code-switching corpus that contain samples of Arabic code-switching was used to produce the Bangor Arabic–English code-switching (BAEC[1]) corpus. The BAEC corpus has 446,081 characters and 45,251 words and focuses on switching between Arabic and English. Second, two Arabic dialect corpora were created: the Saudi dialect corpus (SDC[2]) and the Egyptian dialect corpus (EDC[3]). The SDC has 2,065,867 characters and 210,396 words and contains mixed dialects of Saudi Arabia from social media, such as Facebook and Twitter. The EDC consists of 2,072,165 characters—around 218,149 words—and contains an Egyptian dialect from Facebook. Third, the Non-Authentic Hadith (NAH[4]) corpus is a corpus containing Arabic Hadith from lesser-known Hadith books, providing a new resource for Hadith community research. It consists of 1,621,423 words from 15 non-famous Hadith books (see Chapter 3).
2. Effective detection of code-switching in Arabic text has been achieved using the PPM compression method. The experimental results showed that this method was very effective,

[1] https://github.com/TaghreedT/BAEC
[2] https://github.com/TaghreedT/SDC
[3] https://github.com/TaghreedT/EDC
[4] https://github.com/TaghreedT/NAH-Corpus

reporting an accuracy of 99.8% when detecting of code-switching between the Egyptian dialect and English (see Chapter 4).

3. Effective Hadith components categorisation using the DL methods—the long short-term memory (LSTM), convolutional neural network (CNN), CNN_LSTM—have been achieved when applied to Arabic Hadith text. The experimental results showed that these methods were very effective compared with PPM, SVM, NB and decision tree (DT) classifiers (see Chapter 5).

4. Significant results for character-based segmentation of Isnad and Matan have been achieved using the PPM compression method. The experimental results showed that this method was very effective, reporting an accuracy of 92.78% (see Chapter 6).

5. Effective detection Hadith authenticity has been achieved using the PPM compression method and DL methods, which have not been previously used. The experimental results showed that these methods were very effective compared with the most recent method used, that is, the CML method. Also, the Isnad was found to be the part of a Hadith that resulted in the most effective automatic determination of authenticity. In addition, Matan can be used to judge Hadiths with an accuracy of up to 85% (see Chapter 6).

1.4 Publications

During the current research, several publications have been produced in the fields of Computer Science and Computational Linguistics:

1. Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The abstract was accepted for a presentation at The International Corpus Linguistics Conference 2019.

   Tarmom T; Teahan W; Atwell E; Alsalka M (2019). Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The International Corpus Linguistics Conference 2019. `https://eprints.whiterose.ac.uk/162940/1/tarmom19clV3.pdf`

2. Compression versus Traditional Machine Learning Classifiers to Detect Code-Switching

in Varieties and Dialects: Arabic as a Case Study. This paper was published in *Natural Language Engineering.*

Tarmom T; Teahan W; Atwell E; Alsalka M (2020). Compression versus Traditional Machine Learning Classifiers to Detect Code-Switching in Varieties and Dialects: Arabic as a Case Study. Natural Language Engineering, pp. 1–14. `https://doi.org/10.1017/S135132492000011X`

3. Non-authentic Hadith Corpus: Design and Methodology. This paper was published in *International Journal on Islamic Applications in Computer Science And Technology.*

   Tarmom T; Atwell E; Alsalka MA (2020). Non-authentic Hadith Corpus: Design and Methodology. International Journal on Islamic Applications in Computer Science And Technology. 8(3), pp. 13–19 `https://eprints.whiterose.ac.uk/155642/`

4. Automatic Hadith Segmentation using PPM Compression. This paper was published in the *Proceedings of the 17th International Conference on Natural Language Processing* (ICON).

   Tarmom T; Atwell E; Alsalka MA (2020). Automatic Hadith Segmentation using PPM Compression. In Proceedings of the 17th International Conference on Natural Language Processing (ICON), pp. 22–29.). `https://aclanthology.org/2020.icon-main.4.pdf`

5. Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. This paper was published in the *Annual International Conference on Information Management and Big Data.*

   Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham. `https://link.springer.com/chapter/10.1007/978-3-031-04447-2_14`

6. Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Classify Arabic Hadith Text

   Tarmom T, Atwell E, Alsalka MA. 2022. Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Classify Arabic Hadith Text. IMAN'2022 International Conference on Islamic Applications in Computer Science And Technology Proc.

1.5 Thesis Structure

The current thesis begins with Chapter 2, which focuses on the background and related work by exploring previous studies dealing with Arabic text classification; an introduction to the main methods used in the current research, as PPM, CML algorithms and DL algorithms, which have been used for comparison purposes is also given. Next, Chapter 3 discusses the creation of the new Arabic corpora that have been built for the current study, which involves a code-switching corpus (BAEC), general corpora (SDC and EDC) and NAH Corpus. Then, Chapter 4 explores the detection of code-switching in varieties and dialects using the PPM compression scheme and the classical ML classifier,that is, the support vector machine (SVM). Following this, Chapter 5 examines three different methods: DL, compression-based and CML on two different Hadith corpora to investigate the most effective method of classifying Arabic Hadith based on Isnad, Matan and full Hadith. Chapter 6 uses a character-based PPM compression method to automatically segment the Isnad and Matan. Also, it identifies which part of the Hadith (Isnad, Matan or both) is the most effective for automatically detecting authenticity. Chapter 7 concludes the thesis.

Table 1.1: Thesis Structure.

Chapters	Title
Ch. 2	Background and Related Work.
Ch. 3	New Corpora for Arabic.
Ch. 4	Detect Code-switching in Varieties and Dialect.
Ch. 5	Hadith Components Categorisation.
Ch. 6	Hadith Segmentation and Authenticity Classification
Ch. 7	Conclusion and Future Work.

Chapter 2

Background and Related Work

The aim of this chapter is to explore previous studies that have dealt with Arabic text classification. This chapter also presents an overview of the Arabic language. Furthermore, it explains the main methods used in the current research such as PPM, CML algorithms and DL algorithms, which have been used for comparison purposes. Part of this chapter focuses on automatic code-switching detection and the fundamental ideas beyond it, providing an overview of the previous related research in this area.

The rest of the chapter is structured as follows: Section 2.1 provides the background for the Arabic language; Section 2.2 provides the background for the Arabic Hadiths; Section 2.3 describes the code-switching phenomena; Section 2.4 provides an overview of the existing Arabic corpora; Section 2.5 describes several approaches for the automatic classification of Arabic text, such as compression-based approach usin PPM, CML and the DL approach; Section 2.6 demonstrates the confusion matrix used for evaluation purposes; Section 2.7 discusses the previous related work on automatic code-switching detection; Section 2.8 discusses the previous related work on classifying Arabic text; and Section 2.9 discusses the previous related work on Hadith segmentation. Finally Section 2.10 concludes the chapter.

2.1 Arabic Language Background

The Arabic language "العربية" is ranked the fourth most widely used language in the world (Nwesri et al. 2005). According to Boudad et al. (2018), Arabic-speaking there are more than 422 million people who speak Arabic, and it is the official language for 27 countries.

There are three main varieties of Arabic: CA; MSA; and dialectal/ colloquial Arabic (DA). CA is the language of the Quran, Hadiths and some older traditional books. It is considered as an older style of Arabic and, hence, is no longer used. CA is entirely different from MSA in both its vocabulary and spelling. MSA is the most formal language used among modern Arabic-speaking people (Harrat et al. 2015). It is used in formal letters, TV programmes, newspapers and magazines and education.

However, in daily conversations and social media, most Arabic people use dialects rather than using MSA. There are many varieties of Arabic dialects, such as the Gulf dialect, Egyptian dialect, Levantine dialect, Moroccan dialect and Iraqi dialect. These dialects are divided by their geographical area. In fact, Egyptian TV and cinema have spread their dialect to all Arabic countries, making it the most widely understood dialect in the Arabic world. However, it is often difficult for the speakers of two different dialects to understand the other. Hence, they tend to switch between their dialects and MSA or English.

2.1.1 Arabic Encoding Methods

The most popular encoding scheme on the web (such as used on Facebook, Twitter, YouTube and Google) is UTF-8 (*Encoding Usage Distribution on the Entire Internet.* 2022). Figure 2.1 shows that UTF-8 is currently the most popular technology.

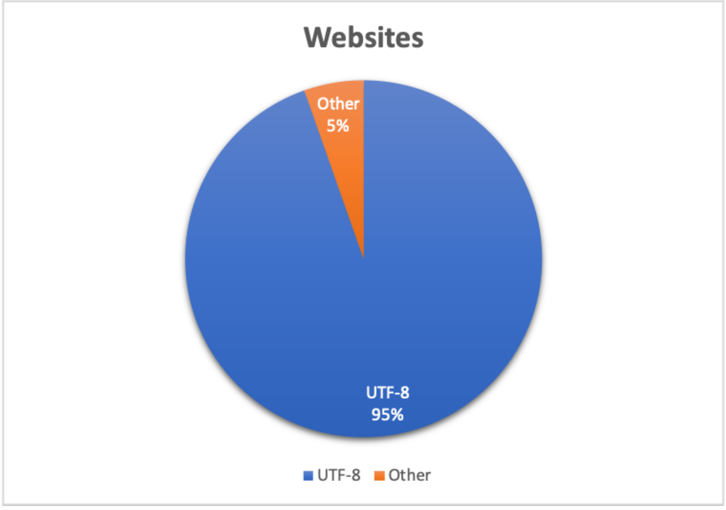


Figure 2.1: UTF-8 is currently the most popular technology on the internet (*Encoding Usage Distribution on the Entire Internet.* 2022)

In UTF-8 encoding, the unit size for Arabic letters is two bytes, and the range for Arabic is between (U+0600) and (U+06FF). UTF-8 is a variable-width encoding because some letters take only one byte and some more (Alkahtani 2015). It is effective for most texts that need more than one byte to encode, such as Japanese and Arabic (Alhawiti 2014).

2.1.2 Arabic Orthography

Unlike English, Arabic is written from right to left, and it does not have upper or lower letters. The alphabet consists of 28 letters, and only three of them are vowels. In addition, Arabic utilises diacritics to describe exact word vowelisation. Arabic people commonly tend to write without diacritics, which makes the text ambiguous for non-native speakers. However, Arabic readers used their semantic knowledge of the language to disambiguate the meaning of terms. For example, the undiacritised word (علم) could have different meanings according to its diacritics: it could be 'flag' (عَلَم) 'teach' (عَلَّم) or it could be 'science' (عِلْمُ). Also, Arabic letters have different shapes depending on their position in the word. All these cause various difficulties for Arabic NLP applications (Farghaly and Shaalan 2009).

2.1.3 Arabic Morphology

The Arabic language is considered to be a rich and complex morphology. There are several morphological aspects for a word in Arabic, including inflection, derivation and agglutination (Boudad et al. 2018).

Inflection Mrphology

Inflection morphology defines the study of the different grammatical categories of a word as a way to describe the same meaning, for example, write, wrote and written. In Arabic, there are several categories for a word inflections, such as for gender (feminine and masculine), number (singular, dual and plural), tense (past and present), person (1st, 2nd and 3rd), voice (active and passive), mood (indicative, imperative, subjunctive) and case (nominative, accusative and genitive) (Boudad et al. 2018). Table 2.1 shows the inflection of the verb كتب.

Table 2.1: The inflection of the verb كتب.

Categories	Past Arabic word		Translated	Present Arabic word		Translated
1st person Singular	كَتَبْت	Katabtu	I wrote	أَكْتُبُ	Aktubu	I write
1st person Plural	كَتَبْنَا	Katabna:	We wrote	نَكْتُبُ	Naktubu	We write
2nd person, Masculine Singular	كَتَبْتَ	Katabta	He wrote	تَكْتُبُ	Taktubu	He write
2nd person Feminine Singular	كَتَبْتِ	Katabti	She wrote	تَكْتُبِين	Taktubin	She write
2nd person, Masculine Singular	كَتَب	katab	He wrote	يكتب	Yktub	He write
2nd person Dual	كَتَبْتُمَا	Katabtuma:	They wrote	تَكْتُبَان	Taktuba:n	They write
2nd person Masculine Plural	كَتَبْتُم	Katabtum	They wrote	تَكْتُبُون	Taktubu:n	They write
2nd person Masculine Plural	كَتَبوا	Katabuu	They wrote	يَكتُبُونَ	Yaktubuuna	They write
2nd person, Feminine Plural	كَتَبْتُنٰ	Katabtunna	They wrote	تَكْتُبنٰ	Taktubunn	They write
2nd person, Feminine Plural	كَتَبنَ	Katabna	They wrote	يَكتُبنَ	Yaktubna	They write

Derivational Morphology

Derivational morphology is the process of changing an existing word to create a new word by adding a derivational suffix or affix; for example, in English the noun ‘writer’ is derived from the verb ‘write’.As other Semitic languages, each Arabic word is based on a root. For example, the verb ‘كتب’ is a root that means write, so if it the letter ‘م’ is added at the beginning, we obtain a new word ‘مكتب’, which is ‘desk’ in English (Boudad et al. 2018).

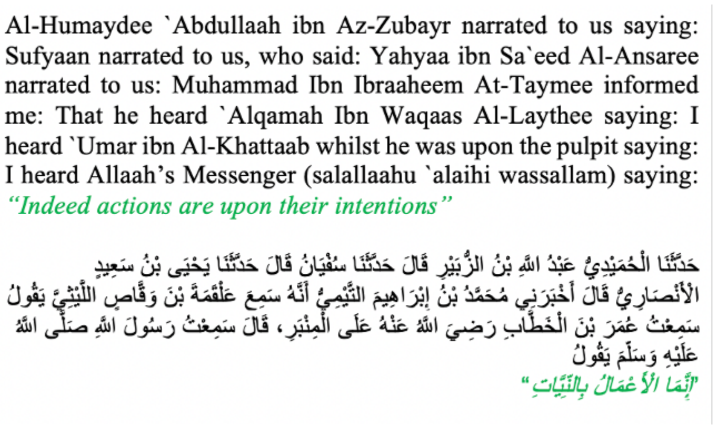
Agglutination Morphology

Agglutination morphology is a linguistic process of attaching a set of affixes into a single word. For example, the Arabic word ‘فَسَيَكْفِيكَهُمُ’ corresponds to the English phrase ‘So will suffice you against them’. This word has five parts (ف + س + يكفي + ك + هم): the ‘ف’ ‘so’, the ‘س’ ‘will’, the ‘يكفي’ ‘suffice’, the ‘ك’ ‘you’ and the ‘هم’ ‘against them’. Because Arabic is an agglutinative language, it has a complex word structure, which causes trouble for Arabic NLP applications (Boudad et al. 2018).

2.2 Arabic Hadith Background

The Prophet Muhammad’s mission started in 610 CE, when he was in a cave outside Mecca, to 632 CE, when he died in Medina. During his mission which was 23-year mission, there was no official writer recording his speech, deeds, orders and his silent approval. However, his companions (Arabic: *sahaba*), memorised the Prophet Muhammad’s legacy, passing this on to others. From generation to generation, his legacy was transmitted in oral/written form (J. A. Brown 2009), until Hadith scholars collected them in books.

Hence, Hadiths—the second source of Islam—refer to any action, saying, order, silent approval or any aspect of the holy Prophet Muhammad’s life or legacy that was delivered through a chain of narrators. Each Hadith has an Isnad—the chain of narrators—and a Matan—the act of the Prophet Muhammad. Figure 2.2 shows an example of a Hadith. This Hadith was written in CA.



Al-Humaydee `Abdullaah ibn Az-Zubayr narrated to us saying: Sufyaan narrated to us, who said: Yahyaa ibn Sa`eed Al-Ansaree narrated to us: Muhammad Ibn Ibraaheem At-Taymee informed me: That he heard `Alqamah Ibn Waqaas Al-Laythee saying: I heard `Umar ibn Al-Khattaab whilst he was upon the pulpit saying: I heard Allaah's Messenger (salallaahu `alaihi wassallam) saying: *"Indeed actions are upon their intentions"*

حَدَّثَنَا الْحُمَيْدِيُّ عَبْدُ اللَّهِ بْنُ الزُّبَيْرِ قَالَ حَدَّثَنَا سُفْيَانُ قَالَ حَدَّثَنَا يَحْيَى بْنُ سَعِيدٍ الْأَنْصَارِيُّ قَالَ أَخْبَرَنِي مُحَمَّدُ بْنُ إِبْرَاهِيمَ التَّيْمِيُّ أَنَّهُ سَمِعَ عَلْقَمَةَ بْنَ وَقَّاصٍ اللَّيْثِيَّ يَقُولُ سَمِعْتُ عُمَرَ بْنَ الْخَطَّابِ رَضِيَ اللَّهُ عَنْهُ عَلَى الْمِنْبَرِ، قَالَ سَمِعْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ

"إِنَّمَا الْأَعْمَالُ بِالنِّيَّاتِ"

Figure 2.2: An example of a Hadith, Isnad in black and Matan in green.

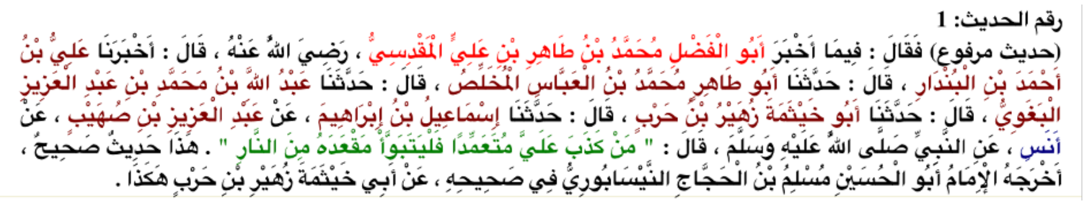
2.2.1 The Importance of Hadiths

The Quran states that the Prophet Muhammad is a role model for Muslims and that he should be obeyed: 'There has certainly been for you in the Messenger of Allah an excellent pattern for anyone whose hope is in Allah and the Last Day and [who] remembers Allah often' (Quran 33:21) 'and obey Allah and His Messenger if you are true believers' (Quran 8:1). Thus, some Muslims imitate the way he lived, such as how he ate and slept. Also, the Muslim community always searches about his legacy and understands it well as a way to support their beliefs, thoughts or political aspects (J. A. Brown 2009). These give the Hadiths importance among Muslims.

In addition, although most ordinances of Islam are mentioned in the Quran in general terms, detailed and vivid explanations are often provided in the Hadiths. For example, the prayer 'الصلاة' is mentioned in the Quran in general terms, while a Hadith specifies what Muslims should do and say; this Hadith explains the time for each prayer and what Muslims should do before and after the prayer.

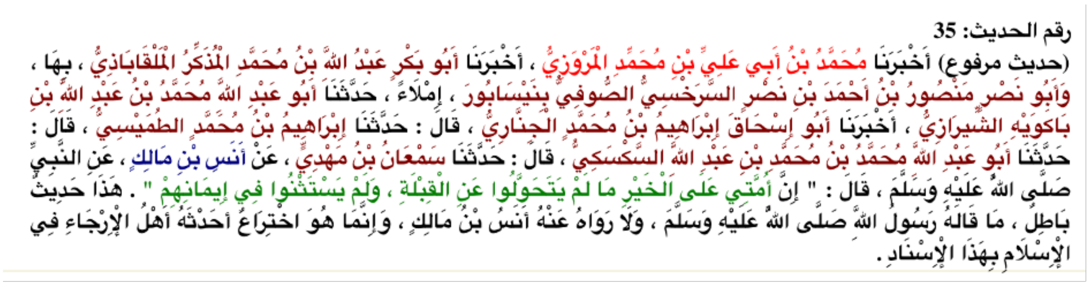
2.2.2 Hadith Authenticity

As described above, that Hadiths are important in all aspects of Muslims' lives. Therefore, Hadith scholars have been interested in studying the validity of the Hadiths. In contrast to the Quran, some Hadiths, which have been handed down over the centuries, have been corrupted by incompetent narrators who have transferred them incorrectly. Hadith scholars have classified these as NAHs. Figures 2.3 and 2.4 show examples of authentic and NAHs.



رقم الحديث: 1
(حديث مرفوع) فَقَالَ : فِيمَا أَخْبَرَ أَبُو الْفَضْلِ مُحَمَّدُ بْنُ طَاهِرِ بْنِ عَلِيٍّ الْمَقْدِسِيُّ ، رَضِيَ اللهُ عَنْهُ ، قَالَ : أَخْبَرَنَا عَلِيُّ بْنُ
أَحْمَدَ بْنِ الْبُنْدَارِ ، قَالَ : حَدَّثَنَا أَبُو طَاهِرٍ مُحَمَّدُ بْنُ الْعَبَّاسِ الْمُخَلِّصُ ، قَالَ : حَدَّثَنَا عَبْدُ اللهِ بْنُ مُحَمَّدِ بْنِ عَبْدِ الْعَزِيزِ
الْبَغَوِيُّ ، قَالَ : حَدَّثَنَا أَبُو خَيْثَمَةَ زُهَيْرُ بْنُ حَرْبٍ ، قَالَ : حَدَّثَنَا إِسْمَاعِيلُ بْنُ إِبْرَاهِيمَ ، عَنْ عَبْدِ الْعَزِيزِ بْنِ صُهَيْبٍ ، عَنْ
أَنَسٍ ، عَنِ النَّبِيِّ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ ، قَالَ : " مَنْ كَذَبَ عَلَيَّ مُتَعَمِّدًا فَلْيَتَبَوَّأْ مَقْعَدَهُ مِنَ النَّارِ " . هَذَا حَدِيثٌ صَحِيحٌ ،
أَخْرَجَهُ الْإِمَامُ أَبُو الْحُسَيْنِ مُسْلِمُ بْنُ الْحَجَّاجِ النَّيْسَابُورِيُّ فِي صَحِيحِهِ ، عَنْ أَبِي خَيْثَمَةَ زُهَيْرِ بْنِ حَرْبٍ هَكَذَا .

Figure 2.3: An example of an authentic Hadith as it appears on *www. islamweb.net.*

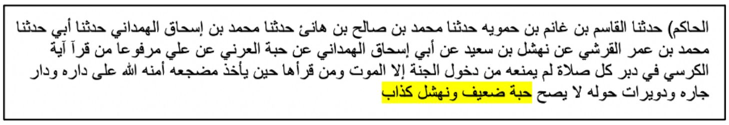


رقم الحديث: 35
(حديث مرفوع) أَخْبَرَنَا مُحَمَّدُ بْنُ أَبِي عَلِيِّ بْنِ مُحَمَّدٍ الْمَرْوَزِيُّ ، أَخْبَرَنَا أَبُو بَكْرٍ عَبْدُ اللَّهِ بْنُ مُحَمَّدٍ الْمُذَكِّرُ الْمُلْقَابَاذِيُّ ، بِهَا ،
وَأَبُو نَصْرٍ مَنْصُورُ بْنُ أَحْمَدَ بْنِ نَصْرٍ السَّرْخَسِيُّ الصُّوفِيُّ بِنَيْسَابُورَ ، إِمْلَاءً ، حَدَّثَنَا أَبُو عَبْدِ اللَّهِ مُحَمَّدُ بْنُ عَبْدِ اللَّهِ بْنِ
بَاكُويَهْ الشِّيرَازِيُّ ، أَخْبَرَنَا أَبُو إِسْحَاقَ إِبْرَاهِيمُ بْنُ مُحَمَّدٍ الْجَنَّارِيُّ ، قَالَ : حَدَّثَنَا إِبْرَاهِيمُ بْنُ مُحَمَّدٍ الطَّمِيْسِيُّ ، قَالَ :
حَدَّثَنَا أَبُو عَبْدِ اللَّهِ مُحَمَّدُ بْنُ مُحَمَّدِ بْنِ عَبْدِ اللَّهِ السَّكْسَكِيُّ ، قَالَ : حَدَّثَنَا سَمْعَانُ بْنُ مَهْدِيٍّ ، عَنْ أَنَسِ بْنِ مَالِكٍ ، عَنِ النَّبِيِّ
صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ ، قَالَ : " إِنَّ أُمَّتِي عَلَى الْخَيْرِ مَا لَمْ يَتَحَوَّلُوا عَنِ الْقِبْلَةِ ، وَلَمْ يَسْتَثْنُوا فِي إِيمَانِهِمْ " . هَذَا حَدِيثٌ
بَاطِلٌ ، مَا قَالَهُ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ ، وَلَا رَوَاهُ عَنْهُ أَنَسُ بْنُ مَالِكٍ ، وَإِنَّمَا هُوَ اخْتِرَاعٌ أَحْدَثَهُ أَهْلُ الْإِرْجَاءِ فِي
الْإِسْلَامِ بِهَذَا الْإِسْنَادِ .

Figure 2.4: An example of a non-authentic Hadith as it appears on *www. islamweb.net.*

However, many forged Hadiths have been circulated not only by incompetent Muslims, but also pious Muslims to encourage their followers to follow their religious and ethical advice. The Isnads exist to clarify Hadith reliability. Today, however, most Muslim scholars cite the Hadiths without providing their Isnads, while, in the early Islamic period, they did not cite Hadiths without mentioning their Isnads (J. A. Brown 2009). This has led to an increasing number of forged Hadiths over time.

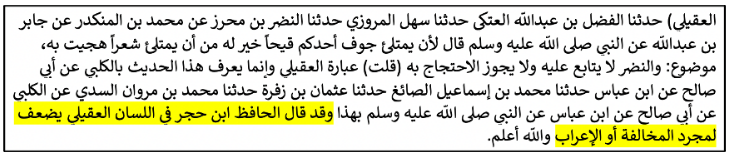
To classify whether the specific Hadith is authentic or non-authentic, Hadith scholars go through the Isnad and Matan. In the Isnad, the narrators must be connected, and the scholars study the states of each narrator, that is, whether they are reliable or accurate (Hakak et al. 2020). Figure 2.5 shows an example of a NAH because of weak and liar narrators (highlighted in yellow).



الحاكم) حدثنا القاسم بن غانم بن حمويه حدثنا محمد بن صالح بن هانئ حدثنا محمد بن إسحاق الهمداني حدثنا أبي حدثنا
محمد بن عمر القرشي عن نهشل بن سعيد عن أبي إسحاق الهمداني عن حبة العرني عن علي مرفوعا من قرآ آية
الكرسي في دبر كل صلاة لم يمنعه من دخول الجنة إلا الموت ومن قرأها حين يأخذ مضجعه أمنه الله على داره ودار
جاره ودويرات حوله لا يصح حبة ضعيف ونهشل كذاب

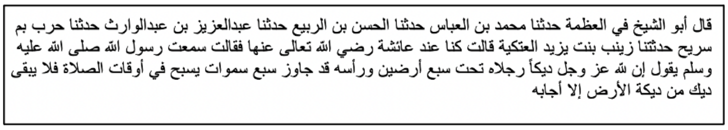
Figure 2.5: An example of a non-authentic Hadith because of weak and liar narrators.

Regarding the Matan, Hadith scholars study whether the Matan contradicts with another authentic Hadith, what is motioned in Quran or with Arabic grammar. Figure 2.6 3 illustrates an example of a Hadith scholar saying that this Hadith has been classified as a NAH because of the contradiction or grammar (highlighted in yellow). Also, in some cases, the Matans contain unacceptable words or expressions that do not reflect the Prophet Muhammad's speech or Muslim beliefs. Figure 2.7 shows an example of a Matan that has an unacceptable explanation about Allah. Figure 2.8 shows the required components of Hadith authentication.



العقيلي) حدثنا الفضل بن عبدالله العتكى حدثنا سهل المروزي حدثنا النضر بن محرز عن محمد بن المنكدر عن جابر
بن عبدالله عن النبي صلى الله عليه وسلم قال لأن يمتلئ جوف أحدكم قيحاً خير له من أن يمتلئ شعراً هجيت به،
موضوع: والنضر لا يتابع عليه ولا يجوز الاحتجاج به (قلت) عبارة العقيلي وإنما يعرف هذا الحديث بالكلبي عن أبي
صالح عن ابن عباس حدثنا محمد بن إسماعيل الصائغ حدثنا عثمان بن زفرة حدثنا محمد بن مروان السدي عن الكلبي
عن أبي صالح عن ابن عباس عن النبي صلى الله عليه وسلم بهذا وقد قال الحافظ ابن حجر في اللسان العقيلي يضعف
لمجرد المخالفة أو الإعراب والله أعلم.

Figure 2.6: An example of a non-authentic Hadith because of a contradiction or the grammar.



قال أبو الشيخ في العظمة حدثنا محمد بن العباس حدثنا الحسن بن الربيع حدثنا عبدالعزيز بن عبدالوارث حدثنا حرب بم
سريح حدثتنا زينب بنت يزيد العتكية قالت كنا عند عائشة رضي الله تعالى عنها فقالت سمعت رسول الله صلى الله عليه
وسلم يقول إن لله عز وجل ديكاً رجلاه تحت سبع أرضين ورأسه قد جاوز سبع سموات يسبح في أوقات الصلاة فلا يبقى
ديك من ديكة الأرض إلا أجابه

Figure 2.7: An example of a Matan that has an unacceptable explanation.

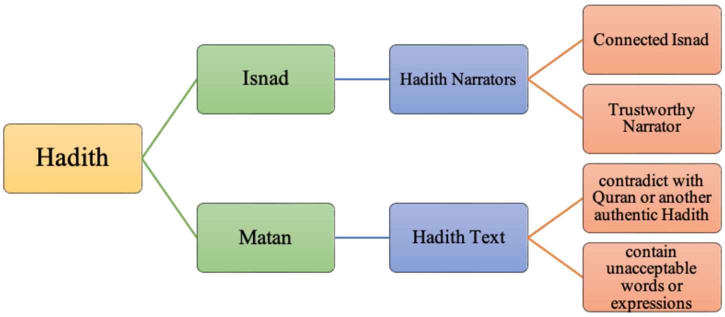

Hadith
Isnad
Matan
Hadith Narrators
Hadith Text
Connected Isnad
Trustworthy Narrator
contradict with Quran or another authentic Hadith
contain unacceptable words or expressions

Figure 2.8: Required components of Hadith authentication (Hakak et al. 2020).

2.3 Code-switching in Arabic

In online communication, code-switching is widespread. Hale (2014) reported that more than 10% of Twitter users wrote in multiple languages; this has also been spotted on other social media platforms (Johnson 2013; Jurgens et al. 2014; Nguyen et al. 2016). However, the occurrence of code-switching in online communication presents a challenge for NLP tools because many of them have been prepared for texts written in only one language.

2.3.1 Code-switching Definitions

In written natural language, the code-switching of text occurs when the author chooses to switch from one language to at least one other. During the past two decades, linguists, sociolinguists and psycholinguists have put forward several definitions for code-switching. Graddol et al. (1996) believed that the scholar's discipline informed the choice of definitions. Most individuals have understood code-switching as occurring when mixing between two (or among several) languages. According to Milroy et al. (1995), this term is 'the alternative use by bilinguals of two or more languages in the same conversation'. Myers-Scotton (2005) defined code-switching similarly as 'the use of two or more languages in the same dialog'. Gumperz (1982) explained code-switching as 'the juxtaposition within the same speech exchange of passages of speech belonging to two different grammatical systems or subsystems'. This latter definition has a wide interpretation and is not restricted to languages, potentially including dialects or formal and informal codes of speech.

2.3.2 Reasons and Motivations for Code-switching

When a bilingual person switches between two languages, this might be because of several reasons or motivations, some of which have been pointed out by Grosjean (1982). For example, code-switching might occur when some bilinguals cannot find an appropriate translation for what they want to say or there is no suitable expression or word in the language being used. Also, some situations, attitudes, emotions and messages generate code-switching. Hidayat (2012) investigated code-switching among Facebook users and reported that 45% of the switching occurred for real lexical needs, 40% for explaining a specific issue, 5% for content clarification, 5% for showing group identity and 5% for quoting somebody else.

Malik (1994) presented 10 motivations for code-switching that could be used to explain the appearance of code-switching in online communication: identification with a group, mood of the speaker, lack of register, lack of facility, amplification and emphasis of a point, semantic significance, habitual expressions, pragmatic reasons, attracting attention and addressing a different audience. Each of these motivations is discussed below.

Identification with a Group

Di Pietro (1977) stated that some Italian emigrants would tell jokes in English with some words in Italian, not only because it was easier, but also to show that they were from the same minority. Moreover, this can be found in Egyptian and Levantine emigrants speakers who have shared identities: they usually end the sentence with phrases such as, مزبوط *(mazboot)* and ماشاء الله *(mashyilhaal)* (Eldin 2014).

Mood of the Speaker

Malik (1994) assumed that code-switching usually takes place when a bilingual individual is angry or tired. Hence, if the speaker is not in their right state of mind, the appropriate expression or word cannot be found in the second language. Often, bilinguals know the expression or word in both languages when they are not upset. For example, Arabic–English bilinguals might speak the following:

[Translated Arabic] [English]
A: ‘eh ilmoshkala? What is wrong?’
B: ‘mosh arfaa, bas I love baba keteer’
Translation: ‘I do not know, but I love my father a lot’ In this instance, A and B are speaking when angry, so they shift to English (Eldin 2014).

Lack of Register

Code-switching also occurs when a speaker does not know the expression in two languages or when they do not have equal proficiency in two languages. Code-switching occurs in specific professions, for instance, in the speech of engineers, lawyers and doctors, because the suitable phrase in Arabic or in any other language save for English might be not available to them (Eldin 2014).

Lack of Facility

According to Malik (1994), code-switching takes place when the proper word cannot be found to carry on the conversation smoothly, as in the following:

[Translated Arabic] [English]
‘eh elkhbar ya man’
Translation: ‘What is up, man?’
In this instance, the speaker clearly did not have an English phrase for *eh elkhbar* (ايه الخبر) (Eldin 2014).

Amplification and Emphasis of a Point

Code-switching can be utilised to confirm a point. Gal (1988) reported that many instances of code-switching occur at the end of a conversation to emphasise a point:

[Spanish] [English]
'Llamé pero no había nadie. I missed him so much!'
Translation: 'I called but there was no one there. I missed him so much!'
In this example, the speaker switches from Spanish to English to emphasise their feeling towards a certain individual (Anderson 2006).

Semantic Significance

Gal (1988) stated that listeners understand code-switching as a sign of the speaker's emotions, attitude and communicative intent. Hence, code-switching is a system that bilinguals use to transfer social information and emotions (Eldin 2014). Also, Crystal (1987) believed that code-switching occurs when a bilingual individual needs to express their feelings or state of mind.

Habitual Expressions

Malik (1994) pointed out that code-switching is often utilised in fixed expressions, such as greetings and partings, invitations, requests and commands and phrases of gratitude, along with in discourse markers such as 'you know' and 'yes', to name a few. For example, consider the following:

[Translated Arabic] [English] [Translated Arabic]
'ana hazoor masr, you know, we hashoof papa we mama'
Translation: 'I will be visiting Egypt, you know, and will see my father and my mother' (Eldin 2014).

Pragmatic Reasons

Sometimes speakers switch between two Sometimes, speakers switch between two languages to add meaning to the conversational context (Malik 1994). Gumperz (1982) considered that moving from one language to another might be to assert varying degrees of a speaker's participation.

Attracting Attention

Malik (1994) showed that in Indian advertisements (both spoken and written), code-switching is utilised to attract the awareness of the listeners/readers. When the reader of English newspapers come across non-English phrases, such as in Hindi or any Indian dialects, the reader's awareness is attracted to it, regardless of their language background (Eldin 2014).

Address a different Audience

Malik (1994) reported that code-switching is also utilised when the speakers intend to declaim individuals coming from different linguistic backgrounds. For instance, the broadcasters in some Egyptian satellite sports channels usually use the national language, which is MSA, but sometimes switch to Gulf and Levantine dialects as well (Eldin 2014).

2.3.3 Code-switching in Arabic

A few studies have researched code-switching in Arabic. The research by Al -Dashti (1998) examined language choices in Kuwait. Here, the most significant age groups that used code-switching were 16–35 and 36–55. Surprisingly, according to Al -Dashti (1998), gender did not have a significant effect on code-switching in Kuwait. Despite this, Kuwaiti women switched to English with children at home more than men, while they switched codes at the same frequency in general outside the home. In contrast, Trudgill (1983), found that gender is one of the most significant factors influencing switching to French in North African Arab countries.

Hussein (1999) studied code-switching among Arabic students, finding that students in the departments of religion and Arabic used code-switching less than students in the other departments. Also, he reported that English phrases, such as ‘OK’, ‘thank you’, ‘yes/no’, ‘sorry’ and ‘please’, were the most frequently used and were used more often than their Arabic equivalents. Warschauer et al. (2002) investigated the use of Arabic–English code-switching in online communications among young Egyptian students; they showed that the participants shifted from Arabic to English to reveal their identity, confirm a point and to show knowledge. S. H. Alfaifi (2013) examined code-switching among bilingual Saudis on Facebook. He found that Saudi women used code-switching on the Facebook social network with friends as a part of their interactions. This could be related to their religion, their cultural experiences, their language environments or to the topics of their communication.

It is clear that code-switching needs more consideration in the context of computer-mediated communication (CMC) in the Arab world, especially in social media, such as Facebook, Instagram, Twitter and WhatsApp (Almesfer 2015).

2.4 Corpus Linguistics

This section is related to Chapter 3 and reviews the existing Arabic corpora that can be used for Arabic NLP tasks.

'Corpus linguistics' can be defined as the study of language through the collection of textual data. It can consist of continuous text from books and websites or can be made up of collections of quotations. Corpora have been compiled for different reasons and purposes. Some existing corpora have been specifically designed for linguistic research, such as the prosody, grammar and discourse patterns of a language (Kennedy 2014). Other corpora have been used for NLP research, such as training and testing materials (Alkahtani and Teahan 2016). Linguistics scholars have found that the manual analysis of huge bodies of text can lead to errors. Thus, computer-based corpora have assisted in the development of an automatic NLP, which has increased the accuracy of most linguistic studies Kennedy (2014).

There are different types of corpora, such as a historical corpus, a parallel corpus, a balanced corpus, a specialised corpus and so on. An **historical corpus** has been collected to study how a language has changed over time, allowing investigators to follow the evolution of the particular linguistic items in a language. The ARCHER[1] (a representative corpus of historical English registers) corpus is an example of this type of corpus; it contains 1.7 million words used between 1650 and 1990. A **parallel corpus** consists of the cotranslations of text in different languages, allowing users to view the examples of a language and their translation equivalents in other languages (Evans 2007). One example of this type of corpus is the parallel corpus for Arabic and English built by Saad Alkahtani, a PhD student at Bangor University; it contains 58 million words from recently published novels and popular Arabic news websites (Alkahtani 2015). A **balanced corpus** consists of the same number of text that mirror a particular kind of text (Lüdeling and Kytö 2008). The Brown corpus, which is the first machine-readable corpus ever produced, is an example of this type of corpus. It contains one million words collected from American English texts and consists of 500 samples of over than 2000 words each (Francis and Kucera 1979). A **specialised corpus** consists of a particular type of text, such as the child language data exchange system (CHILDES) corpus (MacWhinney 2000), which is a corpus that has been collected to study child language acquisition.

[1] `https://www.projects.alc.manchester.ac.uk/archer/`

2.4.1 Existing Arabic Corpora

Corpora are a major factor in NLP, and choosing suitable training corpora is the first step in building language models. We believe that when the training corpus correctly represents the language or dialect under study, the classifier will correctly predict the language.

Some researchers have been prevented from moving forward in their endeavours because building new corpora requires substantial time and effort and because the existing corpora are quite expensive, did not match their purposes and/or are of poor quality. The number of existing Arabic corpora are smaller when compared with the existing English corpora. Therefore, there are few options for Arabic researchers. A. Ahmed et al. (2022) encouraged Arabic NLP researchers to build new, freely available Arabic corpora on areas that are currently unexplored. The Artificial Intelligence research group in the school of computing at the University of Leeds has built several Arabic corpora and made them freely available and can be widely used by Arabic researchers (Atwell 2019; Atwell 2018). In this section, we highlight a list of the existing Arabic corpora that can help Arabic NLP researchers.

The King Saud University corpus of CA (KSUCCA)[2] (Alrabiah et al. 2013) is a freely available balanced corpus consisting of 50 million words. It includes only CA texts from the pre-Islamic era. It was designed to support the linguistics, computational linguistics and historical researchers. It has six different categories: religion, linguistics, literature, science, sociology and biography. This corpus can assist researchers who work in CA. The King Abdulaziz City for Science and Technology (KACST)Arabic corpus consists of over 700 million words (Al -Thubaity 2015). It covers a period of more than 1,500 years, from the pre-Islamic era to when the corpus was built. However, it is only freely available to explore, not to download.

The Bangor Arabic Compression Corpus (BACC) (Alhawiti 2014) is a 31-million-word corpus. It is considered to be the first Arabic corpus built for compression purposes. It was collected from many sources such as books, magazines and websites and includes different genres, such as religion, history, sports and so on. Both MSA and CA can be found in this corpus. Hence, it could be used for training MSA. Unfortunately, there is no way to access the BACC.

The Arabic corpus from the web (arWaC)[3] is a 174-million-word corpus built by Serge Sharoff using Arabic web domains. The international corpus of Arabic (ICA), built by Alansary and

[2] https://sourceforge.net/projects/ksucca-corpus/

[3] https://www.sketchengine.eu/arabic-web-corpus-wac/

Nagi (2014), is an 80-million-word corpus and is considered to be the first international corpus utilising Arabic. It includes only MSA text from all around the Arab world and from many sources, such as newspapers, web articles, books and so on. It includes different genres, such as sports, religion, the arts and so on. Unfortunately, we did not find a way to access this corpus.

Alshutayri and Atwell (2018) reported that social media is a useful source for collecting an Arabic dialects corpus and built an annotated corpus from Twitter, Facebook and online newspapers. Their corpus covers five main Arabic dialects: Gulf, Iraqi, Egyptian, Levantine and North African.

El -Haj (2020) built the first freely available Arabic song lyrics corpus called Habibi. This multidialect corpus consists of over 3.5 million words from over 30,000 Arabic song lyrics. The annotated dialectal Arabic corpus (Twit15DA) (Althobaiti 2021) is another manually annotated corpus built from Twitter, containing 311,785 tweets, that is, around 3,858,459 words in total. It covers 15 Arabic dialects, and each tweet has been labelled by two annotators. This corpus could be used for training Arabic dialects, but there is currently no way to access it.

In this review, we did not find any freely available Arabic code-switching corpus.This gave us a motivation to build an Arabic code-switching corpus and make it freely available for Arabic NLP research.

2.4.2 Existing Arabic Hadith Corpora

The existing Arabic corpora largely focus on Arabic dialects and MSA (Belinkov, Magidow, Romanov, et al. 2016). Because Hadith scripts are considered CA, we are only interested in CA corpora that include Hadiths. There are relatively few Arabic corpora that incorporate Hadith books, such as the KSUCCA (Alrabiah et al. 2013) and KACST Arabic corpus (Al -Thubaity 2015).

Another Arabic corpus that has Hadith books is the Shamela corpus, a large-scale, Arabic historical corpus that contains of a billion words. It was built from the Al-Maktaba Al-Shamela website [4] (Belinkov, Magidow, Romanov, et al. 2016). The Open Islamicate Texts Initiative (OpenITI) is the Arabic part of a project aiming to collect Persian, Arabic and other language texts. It is considered the largest freely available Arabic corpus, comprising of 1,537 million words (Belinkov, Magidow, Barrón-Cedeño, et al. 2018).

[4] `https://shamela.ws/`

There are other Arabic corpora that just focus on Arabic Hadith texts, such as the Sunnah Arabic corpus, which consists of 144,000 words from the *Riyadu Assalihin* ‘رياض الصالحين’ Hadith book (Alosaimy and Atwell 2017). Mahmood et al. (2018) built the multilingual Hadith corpus, which includes Hadiths in different languages, such as Arabic, English and Urdu; they extracted Hadiths from canonical books (major Hadith books that are widely accepted as authentic). In addition, the Leeds and King Saud University (LK) Hadith corpus was recently published to enrich Islamic Hadith resources; it consists of 39,038 annotated Hadiths from the six canonical Hadith books. Much like the previous corpus, it includes Hadiths in different languages, such as Arabic and English (Altammami et al. 2020).

After reviewing the existing Arabic Hadith corpora, we found that most concentrate on extracting text from the canonical Hadith books ‘الصحاح الستة’. Hence, there is a shortage of research on lesser-known Hadith books that have no clear structure. The main motivation for building the NAH corpus was to fill this gap in the research.

2.5 Approaches to the Automatic Classification of Arabic Text

The automatic classification of Arabic text has been achieved using various approaches; currently, classification rates of up to 99% can be reached, even with minimal input. This section briefly reviews the main approaches that have been used, such as n-grams, compression-based language algorithms, CML algorithms (e.g., SVM, NB and k-NN) and DL algorithms (e.g., CNN, recurrent neural network (RNN) and LSTM).

2.5.1 Statistical Language Modelling Techniques

Many applications have successfully used language modelling approaches, such as lexical substitution (Yu, Wu, et al. 2010), grammar error correction (Wu et al. 2010), code-switching language processing (Yu, He, and Chien 2012) (Yu et al., 2012) and text classification. This section reviews some of the features of language modelling in more detail.

A ‘statistical language model’ can be defined as a likelihood distribution P(s) over all possible sentences. For natural language, if the sequence consists of words $(w_1 w_2 \ldots . w_{n-1} w_n)$, the probability for that sequence could be written without a loss of generality as follows:

$$P(s) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2).\dots.P(w_n|w_1w_2.\dots..w_{n-1})) \tag{2.1}$$

(P. F. Brown et al. 1990). The aim of statistical language modelling is to quantify the probability of a given word sequence and identify regularities in a natural language (Rosenfeld 2000).

N-grams

N-grams have been used to classify Arabic text (Khreisat 2006; Khreisat 2009). An *n*-gram is a word slice consisting of length *n* produced from a longer string (Cavnar and Trenkle 1994). The idea behind an n-gram is to divide every string of text into small parts with a maximum length of *n*, count the frequency of the appearance of each n-gram and discard all rarely occurring n-grams. More simply put, it uses the previous *n*-1 words to predict the next word (Teahan 2000):

$$P(w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2).\dots.P(w_n|w_1w_{2-1}.\dots..w_{n-1}) \tag{2.2}$$

A unigram language model uses n=1. Thus, $P(w_i)$ uses the probability distributions of individual words, while a bigram language model uses n=2, and $P(w_i|w_{i-1})$ estimates the probabilities from two-word sequences. A trigram language model uses n=3. $P(w_i|w_{i-2}, w_{i-1})$ estimates the probabilities from three-word sequences.

Khreisat (2006) used trigrams for Arabic text classification. She collected her corpus from online Arabic newspapers and included four categories of text: economy, technology, sports and weather. A total of 40% of this corpus was used as a training set, while 60% was used as a test set. For the preprocessing stage, she removed all diacritics, stop words, nonletters and punctuation marks. Two measures were used to compute precision and recall: the Manhattan distance and Dice coefficient. As such, these two measures were used to compare the performance of the trigram technique. Khreisat's experiments revealed that n-gram text classification using the Dice coefficient had a better classification result than a classification using the Manhattan distance.

Compression-based Approach with PPM

A compression-based approach is another interesting means of automatically classifying Arabic text. This method adopts the PPM text compression algorithm. It is a character-based model that predicts an upcoming symbol by using previous symbols with a fixed context. Every possible upcoming symbol is assigned a probability based on the frequency of previous occurrences. If a symbol has not been seen before in a particular context, the method will 'escape' to another lower-order context to predict the symbol. This 'escape method' is used to combine the predictions of all character contexts (Cleary and Witten 1984).

Different variants of PPM have been created to give better compression results, such as PPMC (Moffat 1990) and PPMD (Howard 1993). Howard (1993), who invented PPMD, showed that PPMD gives can give better results for text compression than PPMC. Equation (2.3) defines how PPMD estimates the probability P for the next symbol $\emptyset$:

$$P(\emptyset) = \frac{2C_d(\emptyset) - 1}{2T_d} \tag{2.3}$$

where d is the coding order, T_d indicates how many times the current context, in total, has existed, and $C_d(\emptyset)$ is the total number of instances for the symbol $\emptyset$ in the current context. Equation (2.4) defines how PPMD estimates the escape probability e:

$$e = \frac{t_d}{2T_d} \tag{2.4}$$

where t_d represents how many times that a unique character has existed following the current context.

Table 2.2 describes how PPM handles the string 'PXYZXY' with an order k=2. For illustration purposes, two has been chosen as the model's maximum order. In order two, if the symbol 'Z' follows the context 'PXYZXY', its probability will be $\frac{1}{2}$ because it has been found before (XY →Z). The encoding of the symbol 'Z' requires $-log(\frac{1}{2}) = 1\ bit$.

If the symbol 'T' follows the context 'PXYZXY', an escape probability of $\frac{1}{2}$ will be arithmetically encoded because it has not been found after ‘XY’ in order two. Then, the PPM algorithm will move to the lower order, which is order one. In order one, because the symbol ‘T’ has not been found after the symbol ‘Y’, an escape probability of $\frac{1}{2}$ will be encoded. Then, this will be repeated in order zero, and an escape probability of $\frac{4}{10}$ will be encoded because the symbol ‘T’ has not been found in order zero. Finally, the algorithm will move to order −1. In this order, all symbols are found, and the probability will be $\frac{1}{|A|}$, where A = 256 (the alphabet size for ASCII), so its probability will be $\frac{1}{256}$. The encoding of the symbol ‘T’ requires $-log(\frac{1}{2} \times \frac{1}{2} \times \frac{4}{10} \times \frac{1}{256} = 11.32\ bits$.

Table 2.2: Handling the string ‘PXYZXY’ using PPM with order 2.

Order k=2			*Order k=1*			*Order k=0*			Order k=-1		
Prediction	c	p	Prediction	c	p	Prediction	c	p	Prediction	c	p
$PX \rightarrow Y$	1	$\frac{1}{2}$	$P \rightarrow X$	1	$\frac{1}{2}$	P	1	$\frac{1}{10}$	A	1	$\frac{1}{\vert A\vert}$
$\rightarrow Esc$	1	$\frac{1}{2}$	$\rightarrow Esc$	1	$\frac{1}{2}$	X	2	$\frac{2}{10}$			
$XY \rightarrow Z$	1	$\frac{1}{2}$	$X \rightarrow Y$	2	$\frac{2}{3}$	Y	2	$\frac{2}{10}$			
$\rightarrow Esc$	1	$\frac{1}{2}$	$\rightarrow Esc$	1	$\frac{1}{3}$	Z	1	$\frac{1}{10}$			
$YZ \rightarrow X$	1	$\frac{1}{2}$	$Y \rightarrow Z$	1	$\frac{1}{2}$	$\rightarrow Esc$	4	$\frac{4}{10}$			
$\rightarrow Esc$	1	$\frac{1}{2}$	$\rightarrow Esc$	1	$\frac{1}{2}$						
$ZX \rightarrow Y$	1	$\frac{1}{2}$	$Z \rightarrow X$	1	$\frac{1}{2}$						
$\rightarrow Esc$	1	$\frac{1}{2}$	$\rightarrow Esc$	1	$\frac{1}{2}$						

Teahan (2000) used PPM to solve several NLP problems, such as segmenting words in both Chinese and English texts, here achieving an accuracy of 99%. This method has also been used to classify text by genre (such as sports, politics or religion) (Teahan 2000). Teahan (2000) reported that using a character-based model required fewer training documents than other methods. In addition, Teahan (2018) concluded that text classification using the PPM character-based compression approach outperformed feature-based approaches, such as NB and SVM. Alkhazi and Teahan (2017) used the PPM classifier to classify MSA and CA and obtained an accuracy rate of 95.5%.

Tawa is a compression-based toolkit that adopts the PPM algorithm. It consists of nine main applications, such as `classify`,`codelength`, `train`, `markup`,and `segment`(Teahan 2018). The current study concentrates on three applications provided by the Tawa toolkit: building models, text segmentation and classification.

2.5.2 Classical Machine Learning Algorithms

Elhassan and M. Ahmed (2015) reviewed the most common CML algorithms that have been used in Arabic text classification: naïve Bayes (NB), SVM and DT. They suggested that three general phases should be included in the process of classifying Arabic text. In the data pre-processing phase, the text should be cleansed of unnecessary words, such as non-Arabic words, stop words, punctuation marks and numbers. Stemming techniques should be used to extract the word root. In the text classification phase, different classifiers should be trained and the results compared to choose the best classifier. Finally, in the evaluation phase, different measures can be used to evaluate the classifier, such as F1, precision and recall. Furthermore, the researchers summarised the classifiers that have been used for different studies and measures that have been used in the evaluation phase (see Table 2.3).

Support Vector Machine

SVM is a supervised CML algorithm used for regression analysis and classification. It has been applied to different NLP problems, such as part-of-speech tagging, information extraction and so on. On unseen data, the SVM classifier has a better generalisation capability than other classifiers (Y. Li et al. 2009). One disadvantage of SVM is that it is slow, especially when applied to a very large classification problem. The enhanced algorithm for SVM, sequential minimal optimisation (SMO), solves SVM problems by dividing a big quadratic programming (QP) problem into a chain of smaller QP problems; this leads to improved results and computation time (Platt 1998). In Weka, sequential minimal optimisation is called ‘SMO’. This SMO version of SVM in Weka has been shown to be effective for other Arabic text classifications (Alshutayri, Atwell, et al. 2016).

Naïve Bayes

NB has been used in a wide variety of NLP tasks and is a supervised CML algorithm based on the Bayes’ theorem. Suppose that there are k classes, $C_1, C_2, \ldots..C_k$. Given a sample A that has no class label, the NB classifier will label it as the class with the highest a posteriori probability (Mitchell 1997), here conditioned on A.

The NB classifier predicts the test example t belongs to the class C, as follows:

$$P(C|t) = \frac{P(C).P(t|C)}{P(t)} \tag{2.5}$$

where $P(C)$ is computed from the number of documents in the category divided by the number of documents in all categories. $P(t)$ is the probability of a test document. $P(t|C)$ is the probability of the test document given the class and is calculated as follows:

$$P(t|C) = \prod_i P(word_i|C) \tag{2.6}$$

This can be rewritten as follows:

$$P(C|t) = P(t) \prod_i P(word_i|C) \tag{2.7}$$

where $P(word_i|C)$ is the probability that a given word occurs in all documents of class C, which is computed as follows:

$$P(word_i|C) = \frac{W_{ct} + 1}{N_c + |V|} \tag{2.8}$$

where W_{ct} is the number of times that the word occurs in class C, $N_c$is the total number of words in class C, and a V is the size of the vocabulary. Finally, to avoid zero probability, 1 has been added to the W_{ct}.

Decision Tree

A DT is a supervised CML algorithm that can be used for text classification. Each tree has a root node, inner nodes and terminal nodes. The root and inner nodes represent the decision stages; these are known as nonterminal nodes. The terminal nodes are the final classification. Also, each tree has layers that consist of a set of nodes, and all have same distance from the root (Swain and Hauska 1977). Each node has a set of categories to be classified, and each subset represents a value that the node can take. Figure 2.9 shows an example of a DT.

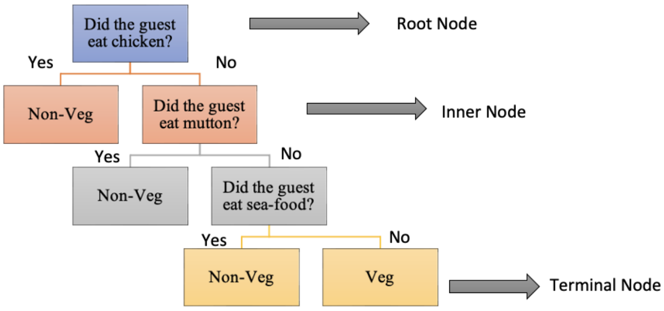

Did the guest eat chicken?
Root Node
Yes
No
Non-Veg
Did the guest eat mutton?
Inner Node
Yes
No
Non-Veg
Did the guest eat sea-food?
Yes
No
Non-Veg
Veg
Terminal Node

Figure 2.9: An example of a decicion Tree.

Table 2.3: : Summary of the classifiers that have been used in Arabic text classification

Reference	Classifiers Used	Accuracy Measure	Best Classifier	Accuracy
(Mesleh 2008)	SVM/NB/k-NN	Precision, recall, F1.	SVM	90%
(Syiam et al. 2006)	k-NN/Rocchio	Precision, recall	Rocchio	98%
(Ababneh et al. 2014)	Cosine/Jaccard/ Dice	Micro recall	Cosine	95%
(Khorsheed and Al-Thubaity 2013)	SVM/C5.0	Accuracy	C5.0	78%
(Al -Shargabi et al. 2011)	SMO/ NB/ J48 (Decision-Tree)	Percentage split method.	SMO	96%
(Gharib et al. 2009)	SVM/NB/k-NN/ Rocchio	Leave one method	SVM	90%
(Bawaneh et al. 2008)	k-NN/NB	K-fold cross validation	k-NN	84%
(Hmeidi et al. 2008)	k-NN/SVM	Recall, precision and F1	SVM	95%
(Bahassine et al. 2020)	DT/SVM	F1	SVM	90%
(Muaad et al. 2022)	Multinomial NB (MNB)/Bernoulli NB (BNB)/ Stochastic Gradient Descent (SGD)/ Logistic Regression (LR)/ Support vector classifier (SVC)/Linear SVC/ CNN	Accuracy	CNN	98%

2.5.3 Deep Learning Algorithms

Recently, DL models have obtained exceptional results in both speech recognition and computer vision (Amin and Nadeem 2018) because they have the ability to handle large data. DL models such as RNN and CNN can be used for text classification tasks, and their performance are better than CML models (J. Zhang et al. 2018). However, they are still rarely used in NLP tasks, specially in Arabic text.

DL basically consists of three layers: the input layer, hidden layers and output layer. First the input data are read in the input layer and then are fed into the subsequent layers of artificial neurons (hidden layers) for further pre-processing. The neurons in these hidden layers use the weighted inputs and biases to produce an output using the activation functions. The output for the given programme is given by the last layer of neurons, which is the output layer.

Boukil et al. (2018) pointed out that DL algorithms such as CNN are widely used in pattern recognition and image processing. However, they are still rarely used in text classification. Also, they mentioned that for a big dataset, DL algorithms such as CNN can accomplish great performance, more so than classical ML algorithms such as SVM.

X. Zhang et al. (2015) utilised character-level convolutional networks for text classification. Several experiments have been carried out to compare traditional methods such as bag of words, bag of n-grams and their TF-IDF and DL methods, such as word-based convolutional networks, character-level convolutional networks and long–short-term memory (LSTM). X. Zhang et al. (2015) experiments showed that the methods under study performed better using the larger corpus. Also, they mentioned that the most effective method for text classification was character-level convolutional network.

We have found that there is a lack of using DL classifiers in Arabic NLP studies, especially for Arabic Hadith text classification. The current study will seek to address this gap in the research

Convolution Neural Networks (CNNs)

CNNs are one of the most popular deep neural networks; they have been commonly applied to image classification. The list of layers in a CNN classifier uses the simplest method to transform the input volume to output volume. Also, there are few distinct layers in a CNN classifier, and each layer uses a differentiable function to transform the input to the output (Jogin et al. 2018).

CNNs can capture the local features of the text. However, for long sequence of words, CNNs cannot preserve long-term dependencies. CNNs consist of node layers, which are the input layer, one or more hidden layers and an output layer. Each node in one layer is connected to all nodes in the next layer and in the previous layer.

When applying a CNN to a sequence of words $w_1, w_2, \ldots .w_n$, each word in this text is associated with an embedding vector of the dimension d, resulting in word vectors $w_1, w_2, \ldots .w_n \in R^d$. The $d \times n$ matrix is fed into a single convolutional layer where a sliding window of size k is passed over the text. Then, to each window in the sequence, the same convolution filter can be applied.

Considering a sequence of words $W_i, ..., W_{i+k}$ the

$$x_i = [w_i, w_{i+1}, \ldots \ldots, w_{i+k}] \in R^{k \times d} \tag{2.9}$$

is the concatenated vector of the i^{th} window. Then, we can apply a convolutional filter to each window, giving scalar values r_i

$$r_i = g(x_i.u) \in R \tag{2.10}$$

where typically more filters can be applied $u_1, ..., u_l$, which then will be represented as a vector multiplied by a matrix U and with an addition of a bias term b

$$r_i = g(x_i.U + b) \tag{2.11}$$

with

$$r_i \in R^l, x_i \in R^{k \times d}), U \in R^{k.d \times l} \; and \; b \in R^l$$

Figure 2.10 shows an example of a sequence of word convolutions with $k = 2$ and dimensional output $l = 3$.

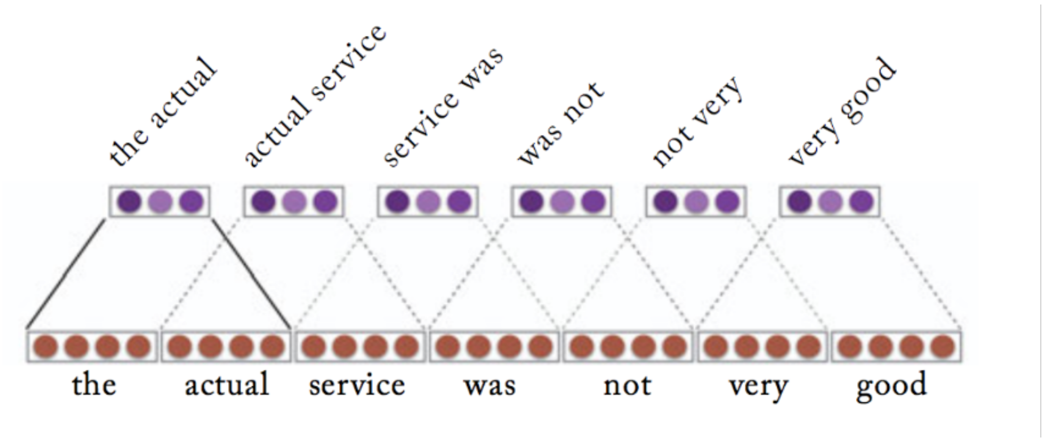


Figure 2.10: An example of a sequence of word convolutions with $k = 2$ and dimensional output $l = 3$ (Goldberg 2017)

Recurrent Neural Network (RNN)

An RNN is a network designed to interpret sequential or temporal information. It can make better predictions by using other data points in a sequence, which is does by taking the input and reusing the activations of previous nodes in the sequence to influence the output. An RNN is very precise in predicting what is coming next because it has internal memory that can remember vital details, such as the input received. Hence, it is the most preferred algorithm for sequential data, such as text, speech and so on.

Long–short-term Memory (LSTM)

One of the limitations of an RNN is that, in each time step, an RNN deals with the previous output of the hidden layer, as well as the current input, without learning long-term dependencies. LSTM is a special type of RNN that addresses this issue (X. Zhang et al. 2015).

In each node state, the LSTM utilises different gates to control the amount of information permitted in it. These gates are the forget gate *f*, the input gate *i*, and the output gate *o*. The equations below provide the LSTM cell and its gates:

1. Input gate:

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i), \tag{2.12}$$

2. Candid memory cell value:

$$\tilde{C_t} = tanh(W_c[x_t, h_{t-1}] + b_c), \tag{2.13}$$

3. Forget gate activation:

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f), \tag{2.14}$$

4. New memory cell value:

$$C_t = i_t * \tilde{C_t} + f_t C_(t-1), \tag{2.15}$$

5. Output gate values:

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o), \tag{2.16}$$

$$h_t = o_t \ tanh(C_t), \tag{2.17}$$

Where W is a weight matrix, x_t is the input to the memory cell at time t, and b is a bias vector. Here, the c indices refer to a cell memory value (Kowsari et al. 2017).

2.6 Confusion Matrix

NLP researchers have often used a confusion matrix to evaluate the performance of their systems. Likewise, we can use it to evaluate the performance of an automatic Arabic text classification system. A confusion matrix is a table that summarises classification and segmentation performance. Table 2.4 presents an example of a three-class confusion matrix.

Table 2.4: A three-class confusion matrix.

	Predicted X	Predicted Y	Predicted Z
Actual X	9	1	2
Actual Y	0	6	1
Actual Z	0	3	8

The first row of Table 2.4 shows that 12 objects belong to class X, 9 of which are correctly predicted as X, one of which is incorrectly predicted as Y and two of which are incorrectly predicted as Z.

The most often utilised case is a two-class confusion matrix, which can present the positive class and negative class for some binary classification problems. In this case, the four cells of this matrix are true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN), as shown in Table 2.5 (Sammut and Webb 2017).

Table 2.5: Confusion matrix for two classes.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

TP is the number of correct predictions that are positive,

FN is the number of incorrect predictions that are negative,

FP is the number of incorrect of predictions that are positive, and

TN is the number of correct predictions that are negative.

From these four outcomes, four measures of classification performance can be defined:

$$\mathit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.18}$$

$$\mathit{Recall} = \frac{TP}{TP + FN} \tag{2.19}$$

$$\mathit{Precision} = \frac{TP}{TP + FP} \tag{2.20}$$

$$\mathit{F\text{-}measure} = \frac{2 \times \mathit{Precision} \times \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} \tag{2.21}$$

2.7 Previous Related Research in Automatic Code-switching Detection

This section is related to Chapter 4 and reviews the previous related research in automatic code-switching detection.

The most common method for auto-detecting code-switching is to build a list of expressions or words for any language and then match these expressions with the model trained from that language. Recently, to make the processing of texts easier, several researchers have concentrated on automatically detecting code-switching at a high level of accuracy at the word rather than document level (Solorio, Blair, et al. 2014) to make the processing of texts easier. NLP tools have been adapted by various studies for code-switched texts, such as by Solorio and Liu (2008) and Peng et al. (2014).

Various techniques have been used to detect code-switching, such as dictionary building, n-gram-based text categorisation, compression-based approaches and so on (which will be discussed below). These techniques provide trade-offs between using small amounts of memory space, having a high degree of accuracy and/or having a faster processing time (Abbas et al. 2010).

Since the mid-1900s, linguists have studied the code-switching phenomenon. In contrast, the NLP community has only recently started to address this phenomenon (Solorio, Blair, et al. 2014). Solorio, Blair, et al. (2014) pointed out that code-switching has posed new research questions, and they expected an increase in NLP research that can address code-switching in the coming years.

Lignos and Marcus (2013) produced a system that outlined both the problems of social media and code-switching in language and detection status. They collected two corpora from Twitter, containing about 6.8 million Spanish tweets and 2.8 million English tweets, to model the two languages. Then, annotation was carried out by using crowdsourcing for tens of thousands of Spanish tweets, around 11% of which included code-switching. This system achieved a 0.936 F-measure in detecting code-switching tweets and a 96.9% word-level accuracy.

Ahmad and Singla (2022) stated that monolingual languages identification could be solved. However, code-switching identification is a complex task and still unsolved. They used two different datasets—the Hindi–English dataset and Urdu–English dataset, which were provided by 14th International Workshop on Semantic Evaluation 2020. The Hindi–English dataset was collected from Facebook posts, comments, WhatsApp chat conversation and tweets consisting of 1700 code-switching sentences. The Urdu-English dataset was collected from Facebook posts and other social media platforms, consisting of 14,000 sentences. Each word in these datasets was annotated with three tags: English (ENG), Hindi (HIN) or other (O). The datasets were split into 80:20 for training and testing purposes. The authors compared three different classifiers: multinomial naïve Bayes (MNB), DT and SVM classifiers (see Table 2.6).

Table 2.6: Classifier comparison results (Ahmad and Singla 2022).

Model	Accuracy (%)	
	Hi-En	Ur-En
MNB	80.06	73.04
DT	82.57	71.19
SVM	83.58	75.79

Dialect detection in Arabic is crucial for almost all NLP tasks and has recently gained strong interest among Arabic NLP researchers. One of the earliest works in this area was by Elfardy and Diab (2012), which addressed the automatic detection of code-switching in Arabic online texts by identifying token-level dialectal words. The authors mentioned that identifying code-switching in written text is a very challenging task because an accompanying speech signal does not exist. They produced a system called AIDA (automatic identification of dialectal Arabic) comprising an MSA morphological analyser, dictionaries, sound-change rules and a set of language models to perform token-level dialect identification. It achieved a token-level F score of 84.9%.

Elfardy, Al-Badrashiny, et al. (2014) modified the existing AIDA system to detect code-switching between the MSA and Egyptian dialect. The new variant used MADAMIRA (Pasha et al. 2014), a tool for disambiguation and morphological analysis for Arabic. Decreasing the possible options for analyses per word was the main advantage of using MADAMIRA. Also, ANERGazet (Arabic named entity recognition gazetteers) (Benajiba et al. 2007) has been used to identify named entities; it consists of 1,545 entries for location names, such as the names of countries, cities and so on, 2,100 entries for people's names and 318 entries for organisation names. Figure 2.11 illustrates the AIDA pipeline after it was modified.

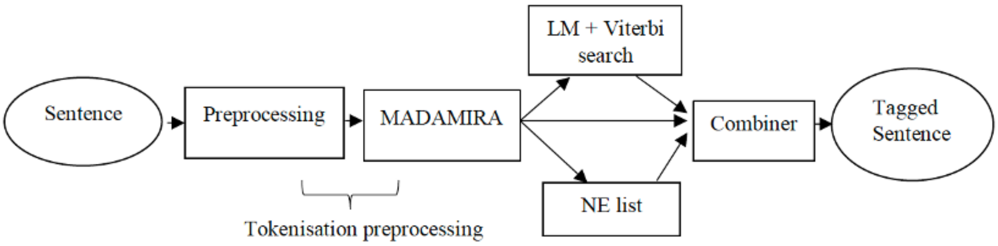


Figure 2.11: The AIDA pipeline with the tokenisation preprocessing scheme (Elfardy, Al-Badrashiny, et al. 2014).

Elfardy, Al-Badrashiny, et al. (2014) used training data from Twitter consisting of 119,326 words and web logs consisting of 8 million words. To evaluate their system, they used the weighted average of the tag F score to rank the overall performance. Three different test sets used to evaluate their system: Test1 consist of 54,732 words, Test2 consist of 32,641 words and Surprise consist of 12,017 words.The system yielded average token-level F scores of 93.6%, 79.9% and 80.1% for the three different tests respectively.

Alkhazi and Teahan (2017) used a PPM character-based compression scheme to segment CA and MSA. It achieved an accuracy of 95.5% and an average F-measure of 0.954 (recall 0.955 and precision 0.958).

2.8 Previous Work on Classifying Arabic Hadiths

This section is related to Chapter 5 and reviews the previous related research in automatic Hadith classification.

Since the mid-1100s, Hadith scholars have attempted to manually classify Arabic Hadiths by topic and authenticity. In contrast, the NLP community has only recently begun to automatically classify Arabic Hadiths. Several algorithms have been used for this task, such as the NB, DT, k-nearest neighbours (k-NN) and SVM. The purpose of this section is to explore previous studies that have focused on building a Hadith corpus and classifying Arabic Hadiths. It also provides an overview of prior related research in this area.

Several studies have focused on automatically classifying Hadiths by topic Naji Al-Kabi et al. (2005); the authors chose eight different topics from the *Sahih Al-Bukhari*, , which is classified as the most authentic Hadith book. It contains 9,082 Hadiths with repetition and 2,602 without repetition. Their system utilised the term frequency, inverse document frequency (TF/IDF) method to perform term weighting. The Isnads and stop words were removed during the filtering process. Next, they converted each word to its root using a stemmer system. They noted that the same Hadith might overlap with two or more topics. In this case, their system displayed the two highest-ranking topics. In the test stage, they selected 80 Hadiths from eight different topics; their system resulted in an accuracy rate of 83.2%.

Alkhatib (2010) compared four different classifiers to automatically classify Hadiths by topic: the NB, SVM, k-nearest neighbour (k-NN) and the Rocchio algorithm. She used 1,500 Hadiths from eight topics in the *Sahih Al-Bukhari*, 1,350 of which were used as a training corpus, while

150 were used as a test corpus. The precision of each classifier was as follows: SVM—63.36%, k-NN—66.55%, NB—66.55% and Rocchio—67.11%. As the classifier with the lowest precision value, SVM was found to be the most accurate algorithm. However, this conclusion seems wrong. Determining the best classifier according to lowest precision value can be seen as a mistake; Rocchio actually appears to be the best classifier because it has the higher precision value at 67.11%.

In their research, Al -Kabi et al. (2014) evaluated the effectiveness of the NB, bagging, SVM and LogiBoost algorithms to classify Arabic Hadiths by topic. They collected their corpus of 793 Hadiths from the *Sahih Al-Bukhari*, choosing the topics of ablutions (*wudu*), fasting, almsgiving (*zakat*), prayers and call to prayers (*adhaan*). They used a total of 474 Hadiths during their training stage. In addition, they removed the diacritics (*tashkil*) and Isnads to enhance the classification results, showing that NB was the most effective algorithm of the four that were tested.

Faidi et al. (2014) compared three different CML classifiers implemented on the Weka toolkit: SMO (the name in Weka for SVM optimised using SMO), NB and J48 (DT) to classify Arabic Hadiths by topics. They built their corpus from the *Sahih Al-Bukhari*, choosing just 795 Hadiths out of 7,031 that were divided into 23 topics. The SMO classifier achieved the highest accuracy rate (57.50%), followed by the NB classifier (48.55%) and the J48 (DT) classifier (44.22%).

Other studies have dealt with automatically classifying Hadiths by authenticity, such as Ghazizadeh et al. (2008). Ghazizadeh et al. (2008) pointed out that, to determine a Hadith's authenticity, two parameters must be used: (1) the reliability and honesty of the Hadith narrators and (2) whether the Hadith was continuous or discrete, as determined by the Isnad. Ghazizadeh et al. (2008) built a fuzzy rule-based system based on these parameters and expert opinion. The system relied on two inference engines. In the first engine, each narrator was ranked according to their reliability and honesty. The second engine used the output from the first step as the input. The second stage of this research produced a Hadith validation rate. To test their system, they used the *Kafi* database, a reliable book of Hadiths. It achieved an accuracy rate of 94%.

Bilal and Mohsin (2012) showed that classifying Hadiths by authenticity is a sensitive and complex task that can only be accomplished by Hadith scholars with intimate knowledge of the large number of rules involved in the process. As a result, the *Muhadith* system was built to facilitate the Hadith classification process. The aims of the *Muhadith* system are to automatically classify Hadiths by imitating Hadith scholars' ability to determine authenticity. It was designed by combining ideas from distributed computing systems, web technologies and Hadith scholars' knowledge. In practice, the user types a Hadith into a web-based interface. The Hadith then passes to the web server, where the user's input is analysed and the required data are extracted. This information is then sent to the fact extractor connected to the database, where the required information is obtained. The results and an explanation of the Hadith classification are then returned to the user.

Siddiqui et al. (2014) used the named entity recognition and classification algorithm to extract the Isnads from Hadiths. Their training corpus was based on text from the *Sahih Al-Bukhari*, and their testing corpus was built from the *Musnad Ahmed*. In their two corpora, each name was tagged as a named entity by a native Arabic speaker. Diacritic marks were removed, and the stemming process was completed. Their research evaluated three classifiers: the NB, the DT and the k-NN (see Table 2.7).

Table 2.7: Overall precision, recall and F1-measures (Siddiqui et al. 2014).

Classifier	Precision	Recall	F1-measure
NB	0.72	0.90	0.80
DT	0.90	0.82	0.86
k-NN	0.83	0.88	0.85

To discover whether the results in Table 2.7 were statistically significant, a paired t-test was used (with 9 degrees of freedom and 5% significance level for the 10-fold cross-validation) to compare the classifiers in a pairwise fashion.Table 2.8 shows the result. The t-test showed that, for the F1-measure, the DT and k-NN were not statistically significant.

Table 2.8: Classifier comparison results (Siddiqui et al. 2014).

Compared Classifiers	Best Classifier for Precision	Best Classifier for Recall	Best Classifier for F1-measure
NB vs DT	DT	NB	DT
NB vs k-NN	k-NN	NB	k-NN
DT vs k-NN	DT	k-NN	None

Different CML algorithms—the SVM, NB and k-NN—were used by Najib et al. (2017) to classify Malay-translated Hadiths based on Isnad. Their test corpus of 100 Hadiths was built from the *Sahih Al-Bukhari* and *Sunan Al-Termizi*. The experiments showed that the SVM classifier obtained the highest accuracy rate, at 82%.

Najiyah et al. (2017) pointed out that non-authentic Hadiths can lead to a misunderstanding of Islamic law. As such, they identified a need to develop an automatic way of classifying authentic and non-authentic Hadiths. They classified Hadiths using expert systems and a DT classifier. First, they created an expert table of Hadiths by interviewing Hadith experts and confirming their findings using a variety of trusted Hadith books. They divided the Hadiths by degree into three groups: (1) *Sahih*, or authentic Hadiths with continuous, trustworthy Isnads and with Matans that did not contradict other authentic Hadiths, (2) *Da'eef*, or Hadiths made weak by non-continuous Isnads, which they then divided into 17 sub-degrees, and (3) *Maudo'*, or fabricated Hadiths created by inauthentic narrators. The degree of a Hadith can be determined by evaluating the Isnad and Matan, as authenticated by Hadith scholars. To evaluate their system, they built a training corpus containing 274 Hadiths and a test corpus containing 72 Hadiths. Their results showed that their classification model could be relied upon to classify *Sahih* Hadith, here with an error rate of only 0.00134%.

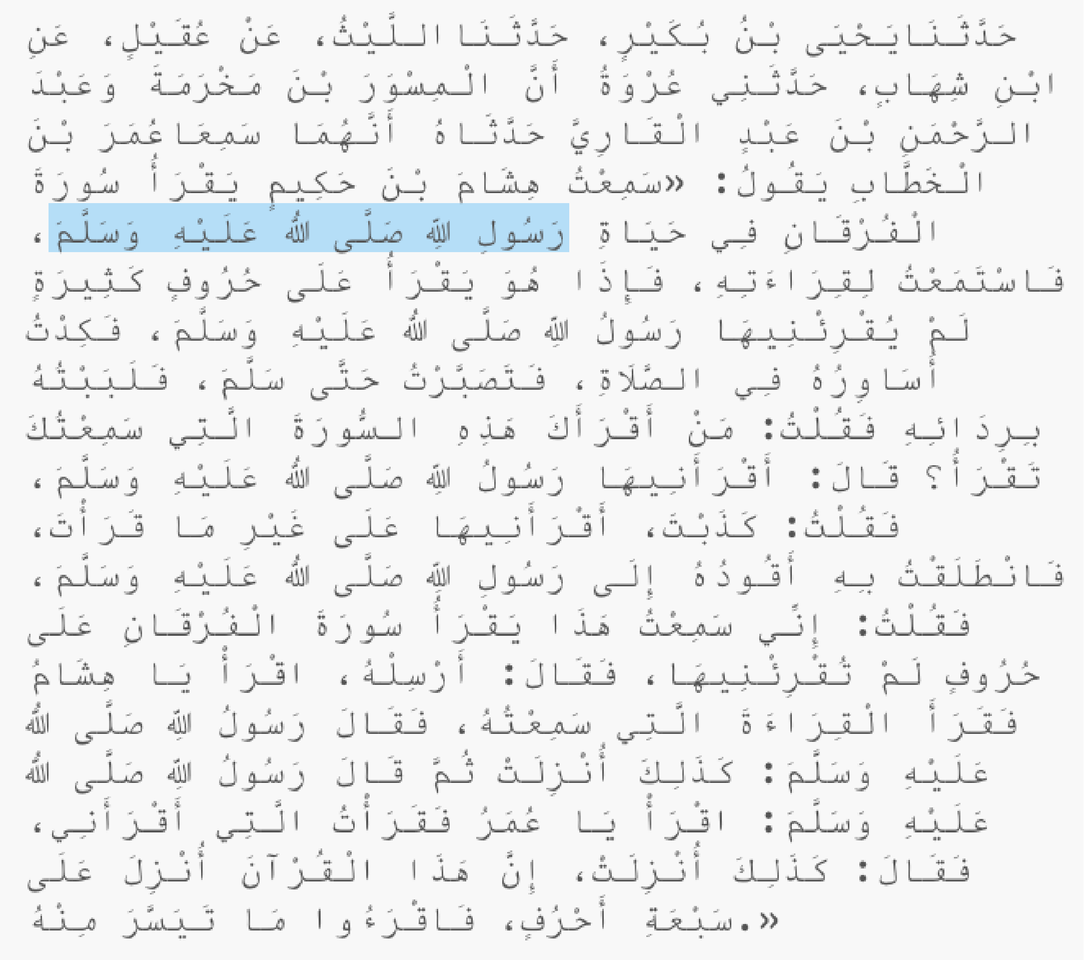
Aldhaln et al. (2012) used the DT algorithm to classify Hadiths according to degree (*Sahih*, *Hasan*, *Da'eef* and *Maudo'*). Their corpus consisted of 999 Hadiths from three different hadith books: *Sahih Al-Bukhari*, *Jami'u Al-Termithi* and *Silsilat Al-Ahadith Al-Dae'ifah w' Al-Mawdhu'ah*. This corpus included both the Hadiths and their attributes, as included in the Hadith books, here as a means of describing their individual degrees. However, some of the Hadiths did not clearly describe these attributes, which resulted in missing values. To solve this problem, the researchers used a missing data detector (MDD). The corpus was divided into two datasets, with 66.7% of the Hadiths comprising the training dataset and 33.3% the test dataset. Their experiments showed that the MDD had a significant effect on the performance of the DT classifier, with the accuracy rising from a rate of 50.15% to 97.59%.

From the previous studies, we noticed that it is important to build NAH corpus and make it available for Hadith researches. This will enrich resources for the Arabic NLP community.

2.9 Previous Work on Hadith Segmentation

This section is related to Chapter 6, and it reviews the previous related research in automatic Hadith segmentation.

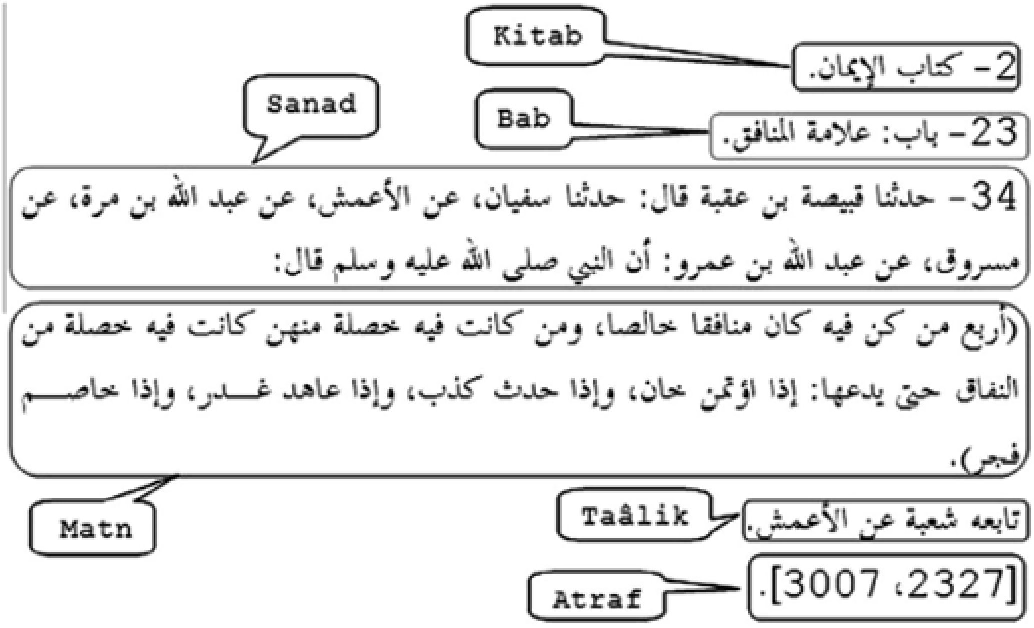
Hadith segmentation can be considered challenging because there are no clear rules specifying where the Matan begins.Figure 2.12 shows that mentioning Prophet Muhammad does not indicate that the Matan will follow.



حَدَّثَنَا يَحْيَى بْنُ بُكَيْرٍ، حَدَّثَنَا اللَّيْثُ، عَنْ عُقَيْلٍ، عَنِ
ابْنِ شِهَابٍ، حَدَّثَنِي عُرْوَةُ أَنَّ الْمِسْوَرَ بْنَ مَخْرَمَةَ وَعَبْدَ
الرَّحْمَنِ بْنَ عَبْدٍ الْقَارِيَّ حَدَّثَاهُ أَنَّهُمَا سَمِعَا عُمَرَ بْنَ
الْخَطَّابِ يَقُولُ: «سَمِعْتُ هِشَامَ بْنَ حَكِيمٍ يَقْرَأُ سُورَةَ
الْفُرْقَانِ فِي حَيَاةِ رَسُولِ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ،
فَاسْتَمَعْتُ لِقِرَاءَتِهِ، فَإِذَا هُوَ يَقْرَأُ عَلَى حُرُوفٍ كَثِيرَةٍ
لَمْ يُقْرِئْنِيهَا رَسُولُ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ، فَكِدْتُ
أُسَاوِرُهُ فِي الصَّلَاةِ، فَتَصَبَّرْتُ حَتَّى سَلَّمَ، فَلَبَبْتُهُ
بِرِدَائِهِ فَقُلْتُ: مَنْ أَقْرَأَكَ هَذِهِ السُّورَةَ الَّتِي سَمِعْتُكَ
تَقْرَأُ؟ قَالَ: أَقْرَأَنِيهَا رَسُولُ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ،
فَقُلْتُ: كَذَبْتَ، أَقْرَأَنِيهَا عَلَى غَيْرِ مَا قَرَأْتَ،
فَانْطَلَقْتُ بِهِ أَقُودُهُ إِلَى رَسُولِ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ،
فَقُلْتُ: إِنِّي سَمِعْتُ هَذَا يَقْرَأُ سُورَةَ الْفُرْقَانِ عَلَى
حُرُوفٍ لَمْ تُقْرِئْنِيهَا، فَقَالَ: أَرْسِلْهُ، اقْرَأْ يَا هِشَامُ
فَقَرَأَ الْقِرَاءَةَ الَّتِي سَمِعْتُهُ، فَقَالَ رَسُولُ اللهِ صَلَّى اللهُ
عَلَيْهِ وَسَلَّمَ: كَذَلِكَ أُنْزِلَتْ ثُمَّ قَالَ رَسُولُ اللهِ صَلَّى اللهُ
عَلَيْهِ وَسَلَّمَ: اقْرَأْ يَا عُمَرُ فَقَرَأْتُ الَّتِي أَقْرَأَنِي،
فَقَالَ: كَذَلِكَ أُنْزِلَتْ، إِنَّ هَذَا الْقُرْآنَ أُنْزِلَ عَلَى
سَبْعَةِ أَحْرُفٍ، فَاقْرَءُوا مَا تَيَسَّرَ مِنْهُ.»

Figure 2.12: An example of a Hadith from *Sahih Al-Bukhari*, showing that mentioning Prophet Mohammad PBH does not indicate that the Matan will follow .

There have been relatively few studies on the segmentation of Hadiths into Isnads and Matans. One study was carried out by Harrag (2014), who developed a finite state transducers-based system to detect the different parts of a Hadith, such as *Title-Bab*, *Num Hadith*, *Sanad* ‘Isnad’, and *Matn* ‘Matan’. The disadvantage of this system is that it was built to depend on the Hadith structure in *Sahih Al-Bukhari* (the most trusted Hadith book), which cannot be used for other Hadith books. Figure 2.13 shows the Hadith structure in the *Sahih Al-Bukhari*. This system achieved a precision of 0.44 for Isnad extraction and 0.61 for Matan extraction.



2- كتاب الإيمان.

23- باب: علامة المنافق.

34- حدثنا قبيصة بن عقبة قال: حدثنا سفيان، عن الأعمش، عن عبد الله بن مرة، عن
مسروق، عن عبد الله بن عمرو: أن النبي صلى الله عليه وسلم قال:

(أربع من كن فيه كان منافقا خالصا، ومن كانت فيه خصلة منهن كانت فيه خصلة من
النفاق حتى يدعها: إذا اؤتمن خان، وإذا حدث كذب، وإذا عاهد غدر، وإذا خاصم
فجر).

تابعه شعبة عن الأعمش.

[2327، 3007].

Figure 2.13: An example of a Hadith structure in *Sahih Al-Bukhari* (Harrag 2014).

Mahmood et al. (2018) selected authentic and reliable Hadith sources such as *Sahih Al-Bukhari*, *Sahih Muslim* English, and *Sunan Abu Dawud.* . Because these books differ in format, structure, length and content, the researchers used different kinds of regular expressions (Regex) for data extraction. However, the Hadith patterns extracted by their system lack detail. The results obtained by their system are summarised in Table 2.9.

Table 2.9: Results of different Hadith books (Mahmood et al. 2018).

Book Name	Precision	Recall	F1 Measure
Sahih Muslim English	96%	91%	93%
Sahih Bukhari English	99%	99%	99%
Sunan Abudawud	100%	100%	100%
Mawta Imam Malik	100%	100%	100%

Maraoui et al. (2018) implemented a segmentation tool to automatically segment Isnads and Matans from each text in *Sahih Al-Bukhari.* First, they analysed the *Sahih Al-Bukhari* corpus and identified the words that distinguish Isnads from Matans. These words were then added to the trigger word dictionary. This tool achieved a precision of 96%.

Altammami et al. (2019) built a Hadith segmenter using n-grams. The *Sahih Al-Bukhari* was selected as a training set, and the testing set was manually extracted from the six canonical

Hadith books. Their results showed that using bi-grams achieved a much higher accuracy (92.5%) than tri-grams (48%). The disadvantage of their segmenter was that it can segment first Isnads and Matans and ignore the others Isnads and Matans if there were more than one.

Most Hadith segmentation and classification research works have used the six famous Hadith books, called the Authentic Six 'الصحاح الستة'. Hence, there is a shortage of research on lesser-known Hadith books, such as the *Fake Pearls of the Non-Authentic Hadiths* 'اللآلئ المصنوعة في الأحاديث الموضوعة'. These books contain a mixture of authentic and non-authentic Hadiths and do not have a clear structure, which makes NLP tasks more complex. Also, character-based text compression methods have not been used in previous Hadith segmentation/classification studies. Our work seeks to fill these gaps in the research.

2.10 Conclusion

We have reviewed several algorithms that can be used to classify Arabic text. Some of these algorithms have been explored by Arabic researchers, while others still need further investigation, specifically using compression-based classifiers and DL classifiers.

We provided background information for the Arabic language, explaining how the Arabic language is rich and complex. Also, we provided the background for the Arabic Hadith, discussing the Hadith importance in the Islamic world and how to determine a Hadith's authenticity. We also reviewed the code-switching phenomena and reasons for it. In addition, we have provided the background for previous work on detection code-switching in Arabic text. We have also reviewed the existing Arabic corpora with an emphasis on Hadith corpus.

This chapter reviewed the approaches to the automatic classification of Arabic text, such as n-grams, compression-based approach with PPM, CML and DL approach. Finally, we set the previous related work on automatic code-switching detection, classifying Arabic text and Hadith segmentation. This review has shown that most of Hadith corpora have been built from the canonical Hadith books, called the Authentic Six 'الصحاح الستة'. Hence, there is a shortage of research on lesser-known Hadith books, such as the *Fake Pearls of the Non-authentic Hadiths* ‘اللآلئ المصنوعة في الأحاديث الموضوعة’.

Chapter 3

Corpora for Arabic

A large collection of textual data in one language or more is called a *corpus* (plural *corpora*). Linguistics scholars found that using computer-based corpora lead to increase the accuracy of most linguistic studies (Kennedy 2014). The use of language computer-based corpora become increasingly important in various NLP tasks, such as classification, segmentation and so on to train models, testing and evaluation.

The first step in building language models is choosing suitable training corpora. As mentioned in the previous Chapter that there are few options of the existing Arabic corpora for Arabic researchers. Therefore, we have built several Arabic corpora for our research and made them freely available to the Arabic research community.

This chapter first discusses the production of dialectal Arabic corpora with aim to test the PPM compression and CML methods on MSA and Arabic dialect before test them on Hadith. It also discusses the production of the NAH corpus in detail.

This chapter includes corpus discerptions from the following papers:

1. *Tarmom T; Teahan W; Atwell E; Alsalka M (2019), Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The International Corpus Linguistics Conference 2019.*

2. *Tarmom T; Teahan W; Atwell E; Alsalka M (2020), Compression versus Traditional Machine Learning Classifiers to Detect Code-Switching in Varieties and Dialects: Arabic as a Case Study. Natural Language Engineering, pp. 1–14.*

3. *Tarmom T; Atwell E; Alsalka MA (2020), Nonauthentic Hadith Corpus: Design and Methodology. International Journal on Islamic Applications in Computer Science And Technology. 8(3), pp. 13–19*

4. *Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.*

The rest of the chapter is structured as follows: section 3.1 discusses the new Arabic corpora that have been built for the current study; section 3.2 describes creating the dialectal Arabic corpora, which involves code-switching corpus and general corpora. It also discusses the methods used to create our corpora; section 3.3 describes creating the NAH corpus, involving the collection process, the quality evaluation of the corpus and the annotation process of the corpus; section 3.4 reports several challenges that we encountered while building our corpora; and finally, section 3.5 concludes the chapter.

3.1 Corpora for Arabic

Most NLP research for the Arabic language focuses on MSA; research in Arabic dialects and CA (such as Hadith) are sparse. In this chapter, the focus is on creating two main different corpora; dialectal Arabic corpora, which include the Bangor Arabic–English Code-switching Corpus (BAEC) corpus, Saudi Dialect Corpus (SDC) and Egyptian Dialect Corpus (EDC). Also, the CA corpora includes the Non-Authentic Hadith (NAH) corpus.

To evaluate a compression-based approach and classical ML classifiers for the automatic detection of code-switching, it was necessary to build a code-switching corpus containing samples of Arabic code-switching and general dialect corpora. Therefore, we created the BAEC corpus for a testing set. The following general dialect corpora have also been created for this research to use as training sets:

- SDC;
- EDC.

In addition, to evaluate different classifiers with respect to the automatic classification of the

Hadith's authenticity, it was necessary to build an Arabic Hadith corpus that contained samples of authentic Hadiths and NAH, which could then be used for training models and testing. We already had access to the LK Hadith corpus (Altammami et al. 2019), which contains just authentic Hadith. Therefore, for a balanced corpus of both positive and negative examples, we created the NAH corpus.

Table 3.1 lists the number of words, number of characters and overall size for each of these corpora.

Table 3.1: Summary of our corpora produced for the current research.

Corpus name	Number of Words	Number of Characters	Corpus size
BAEC	45,251	446,081	436 KB
SDC	210,396	2,065,867	2,018 KB
EDC	218,149	2,072,165	2,024 KB
NAH	1,621,423	9,728,538	14,6 MB

3.2 Dialectal Arabic Corpora

Most Arabic corpora concentrate on MSA, but there has been a shortage in freely available Arabic dialects corpora. This is the main motivation for building new Arabic corpora. Also, for training purposes, we needed a separate corpus for each dialect under research, so we needed further new Arabic corpora for each dialect because these also do not exist. The size of the corpora was another motivation. Our aim was to have corpora with 200,000 words or more each because this was the size recommended by A. Alfaifi (2015). Also, the existing Arabic corpora are quite expensive and/or are of poor quality (Alkahtani 2015). Hence, there was a need to create free Arabic corpora. Therefore, we have created three new Arabic corpora, which took approximately three months to complete. These were the SDC, EDC and BAEC corpus.

3.2.1 Methodology

Different methods were used to create our new corpora. The Facebook Scraper (FS) system helped us automatically extract data from Arabic Facebook pages. The objectives of FS are as follows:

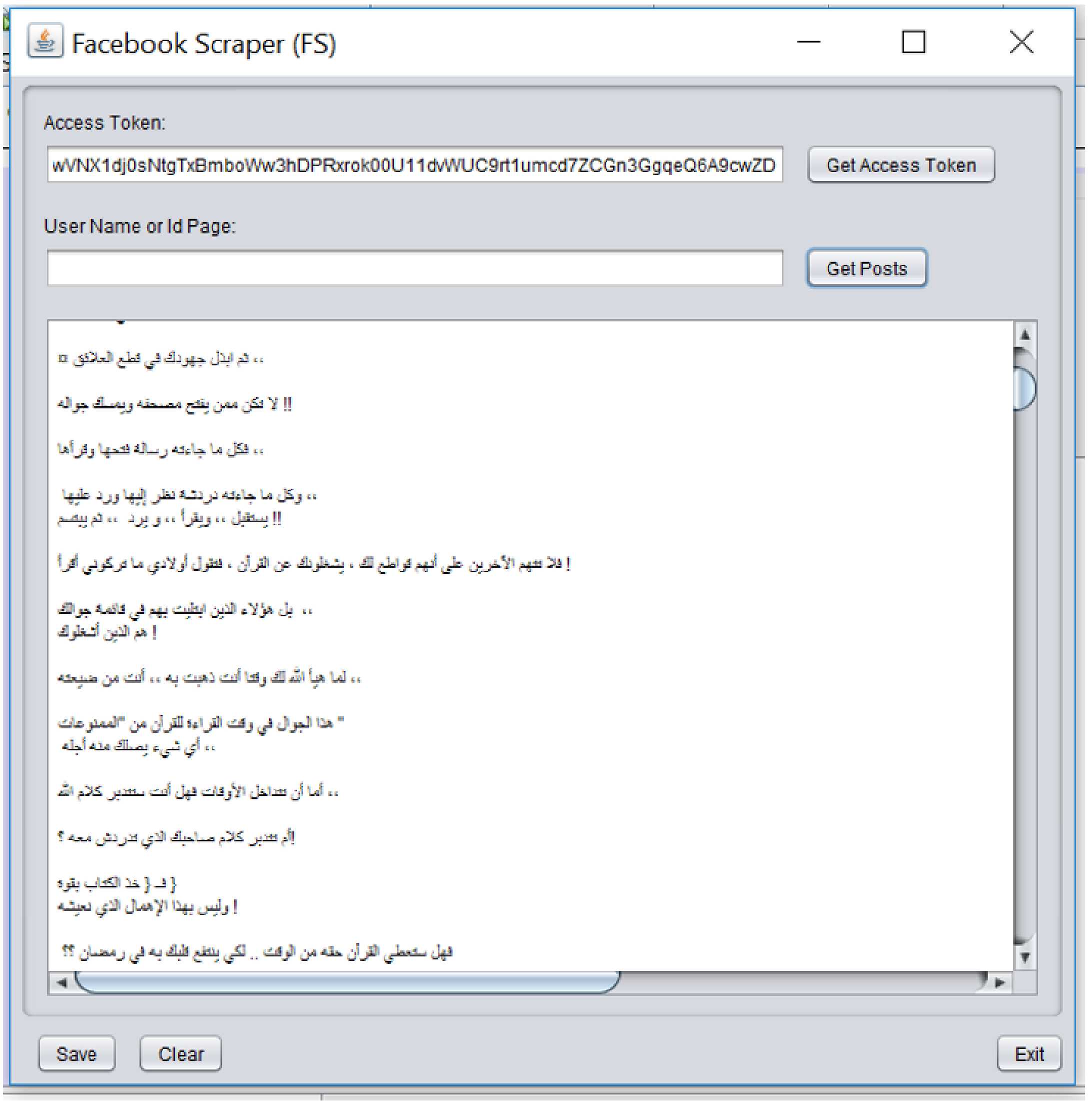
1. To create a system that is effective at scraping pure text from Facebook
2. To create a system that has an easy-to-use interface
3. To create a system that saves time for the user

Extracting data manually takes a long time, so FS helps obtain a large number of posts in the shortest time. The application allows one to collect data from different pages and place them in one text file.

The user generally wants an application that is simple and easy to learn and use. So, the KISS (Keep It Simple, Stupid) principle was taken into consideration when the FS interface was developed.

Java was selected as the development programming language for constructing the programme; it provides various libraries, such as `RestFB` , that help to fetch information from Facebook.

The FS system has been tested many times because it was used to extract posts from many Arabic Facebook pages to build the corpora required for our study. Figure 3.1 below is a screenshot of the FS system.



Access Token: wVNX1dj0sNtgTxBmboWw3hDPRxrok00U11dWWUC9rt1umcd7ZCGn3GgqeQ6A9cwZD

User Name or Id Page:

¤ ثم ابذل جهودك في قطع العلائق ،،

لا تكن ممن يفتح مصحفه ويمسـك جواله !!

فكل ما جاءته رسـالة فتحها وقرأها ،،

وكل ما جاءته دردشـة نظر إليها ورد عليها ،،
يستقبل ،، ويقرأ ،، و يرد ،، ثم يبتسم !!

فلا تتهم الآخرين على أنهم قواطع لك ، يشغلونك عن القرآن ، فتقول أولادي ما تركوني أقرأ !

بل هؤلاء الذين ابقيت بهم في قائمة جوالك ،،
هم الذين أشـغلوك !

لما هيأ الله لك وقتا أنت ذهبت به ،، أنت من ضيعته ،،

هذا الجوال في وقت القراءه للقرآن من "الممنوعات "
أي شـيء يصلك منه أجله ،،

أما أن تتداخل الأوقات فهل أنت سـتتدبر كلام الله ،،

أم تتدبر كلام صـاحبك الذي تدردش معه ؟!

فـ { خذ الكتاب بقوة }
وليس بهذا الإهمال الذي نعيشـه !

فهل سـتعطي القرآن حقه من الوقت .. لكي ينتفع قلبك به في رمضان ؟؟

Figure 3.1: Screenshot of the Facebook Scraper system.

For SDC, we could not collect enough text from Facebook alone because most Saudi Facebook users tend to use non-colloquial Arabic. Hence, we moved to Twitter, the third most popular social network platform in Saudi Arabia (*Saudi Arabia Social Media Statistics 2018 – Official GMI Blog* 2019) (see Figure 3.2). Manual cut-and-paste techniques were used to extract data from Twitter and other websites. For general dialect corpora, extensive cleaning removed emojis, punctuation marks, URLs and non-Arabic words. For the Saudi and Egyptian dialect corpora, we also removed Quranic Arabic and Hadith (the Prophet Mohammad's speech, words and actions) because they are classified as CA.

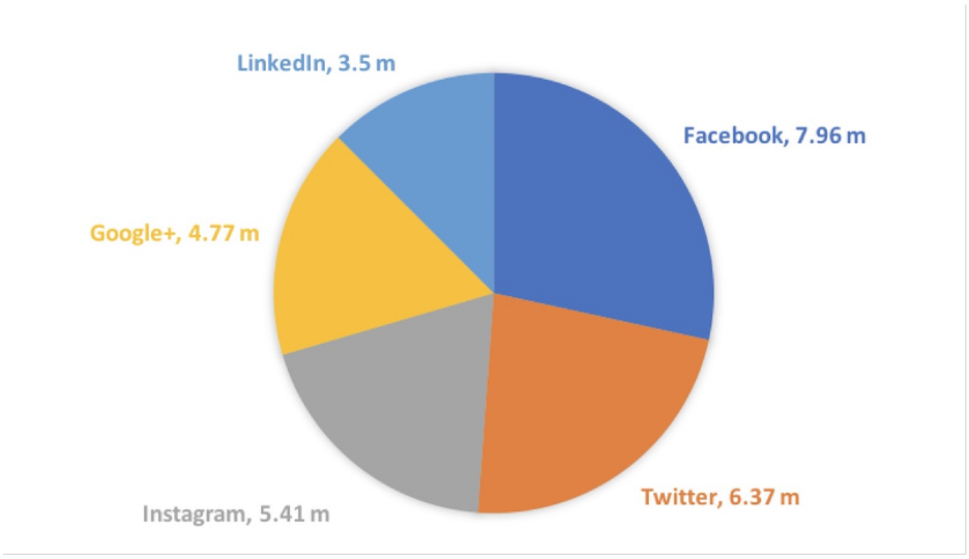


Figure 3.2: Most popular social network platforms in Saudi Arabia 2018 (in millions) (*Saudi Arabia Social Media Statistics 2018 – Official GMI Blog* 2019).

Sampling Method

A judgemental, nonprobability type sampling method was chosen to collect the data. The procedure for collecting a sample is based on personal judgement, so the researcher uses their own experience and knowledge to select a sample (Doyle 2016). When we built our corpora, we looked at the user's location (if available), as well as reading each post and tweet to verify its class (whether written in a Saudi dialect or Egyptian dialect). This required substantial time and effort, taking approximately three months of work.

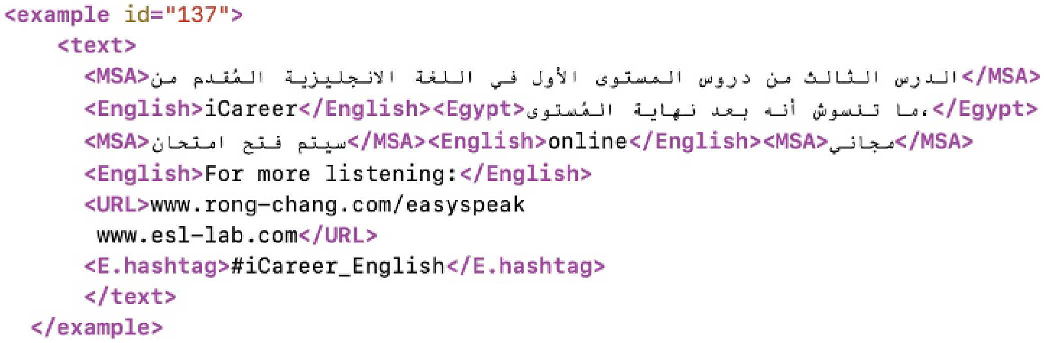
Verifying the Quality of the Tagging

Annotated Arabic dialects are more challenging than MSA because they do not have clear spelling standards and conventions. They have not been commonly written until recently, whereas MSA has official orthographic standards and conventions. Today, Arabic dialects are often used on social media. Two Saudi university researchers with extensive knowledge in Egyptian dialect verified the quality of the tagging. If they disagreed on a particular word, whether in MSA or an Arabic dialect, they looked at the Arabic online dictionary (`www.almaany.com`), which contains all MSA words with their meanings, and came to an agreement.

3.2.2 Bangor Arabic–English Code-switching (BAEC) Corpus

The term '*code-switching corpus*' refers to a body of text consisting of two or more languages under study (Yu, He, Chien, and Tseng 2013). Because Arabic and English are the primary languages in the current research, we built an Arabic–English code-switching corpus. To the best of our knowledge, there are no available Arabic code-switching corpora derived from Facebook, so it was necessary to build a new corpus for our research.

One of the objectives of the current study was to detect code-switching in Arabic text from Facebook. Therefore, we collected our corpus from different Arabic Facebook pages containing code-switching and used it to build the BAEC[1] corpus, which focuses on switching between Arabic and English. It consists of 45,251 words and is 436 KB in size (see Table 3.1). It was collected from different Facebook pages using the FS system (discussed in Section 3.2.1). It includes code-switching between MSA and English; the Saudi dialect and English; and the Egyptian dialect and English. Manually annotated, it has been produced in XML. A sample taken from the BAEC corpus is shown in Figure 3.3. In Figure 3.3, <example id= '137'> is the opening tag for the sample, which has the Id of 137 in this example. This is followed by the actual text between the <text> and </text> tags. Further, <MSA> specifies which of the text is MSA text, <Egypt> specifies the Egyptian dialect, and <English> specifies the English text. Finally, <URL> specifies the URL and <E.hashtag> specifies an English hashtag. Table 3.3 explains each tag used in our corpora.



```
<example id="137">
    <text>
      <MSA>الدرس الثالث من دروس المستوى الأول في اللغة الانجليزية المُقدم من</MSA>
      <English>iCareer</English><Egypt>ما تنسوش أنه بعد نهاية المُستوى،</Egypt>
      <MSA>سيتم فتح امتحان</MSA><English>online</English><MSA>مجاني</MSA>
      <English>For more listening:</English>
      <URL>www.rong-chang.com/easyspeak
       www.esl-lab.com</URL>
      <E.hashtag>#iCareer_English</E.hashtag>
      </text>
  </example>
```

Figure 3.3: A sample from the BAEC corpus.

[1] https://github.com/TaghreedT/BAEC

Table 3.2: Explanation of the tags used in the BAEC corpus.

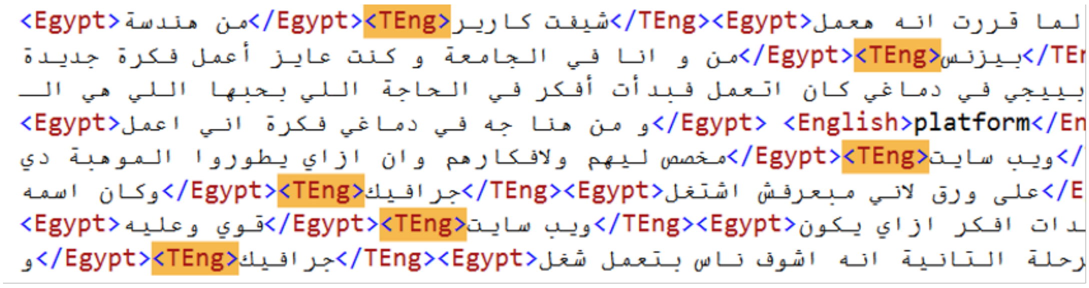
Tag	Explanation
<Saudi>	Saudi dialect
<Egypt>	Egyptian dialect
<English>	English
<MSA>	Modern Standard Arabic
<HQ>	Holy Quran
<Hadith>	Hadith
<MAE>	Mixed Arabic English
<TEng>	Translated Arabic English
<A.hashtag>	Arabic hashtag
<E.hashtag>	English hashtag
<URL>	URL
<email>	Email
<date>	Date
<no.E>	English number
<no.A>	Arabic number
<MISC>	Miscellaneous such as emoji

Corpus Annotation

We used the following rules when we annotated the BAEC corpus: (1) if the sentence was written in Arabic letters and had no Saudi or Egyptian dialect word, we annotated it as <MSA>; (2) if a phrase was written in Arabic letters and contained any Egyptian dialect word, we annotated it as <Egypt> (this rule was also applied to the Saudi dialect); (3) if the word or the phrase was written in English letters, we tagged it as <English>.

We found that tagging the BAEC corpus was much more complex than we had first thought because it had a lot of emojis, URLs, English numbers, Arabic numbers, English hashtags, Arabic hashtags and non-Arabic words.

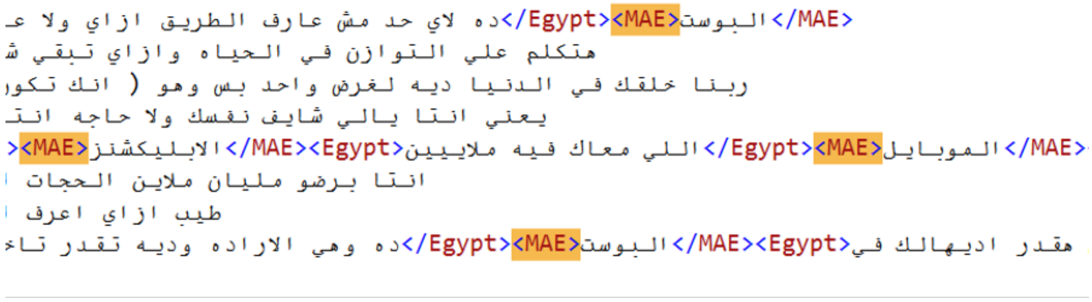
In annotating the BAEC corpus, we found some unforeseen issues. For example, some Arabic Facebook users wrote some English words using Arabic letters, such as منشن 'mention', اونلاين 'online', كورس 'course', شير 'share' and so on. These words have been annotated as translated English <TEng>. Figure 3.4 shows some translated English words written on Facebook.



```
لما قررت انه معمل</TEng><Egypt>شيفت كارير</Egypt><TEng>من هندسة<Egypt>
</TEr بيزنس</Egypt><TEng>من و انا في الجامعة و كنت عايز أعمل فكرة جديدة
بييجي في دماغي كان اتعمل فبدأت أفكر في الحاجة اللي بحبها اللي هي الـــ
</Egypt> <English>platform</En و من هنا جه في دماغي فكرة اني اعمل<Egypt>
</ ويب سايت</Egypt><TEng>مخصص ليهم ولافكارهم وان ازاي يطوروا الموهبة دي
</E على ورق لاني مبعرفش اشتغل</TEng><Egypt>جرافيك</Egypt><TEng>وكان اسمه
ـدات افكر ازاي يكون</TEng><Egypt>ويب سايت</Egypt><TEng>قوي وعليه<Egypt>
رحلة التانية انه اشوف ناس بتعمل شغل</TEng><Egypt>جرافيك</Egypt><TEng>و
```

Figure 3.4: Some examples of translated English words (TEng) found in our corpus.

Also, some users mixed Arabic with English words to produce one mixed word, here based on a normal word such as ‘class’, where they added Arabic letters ال and then wrote الكلاس ‘Alclass’. Another habit was to use Arabic grammar in an English word, for example, making ‘class’ plural by using Arabic grammar rules for pluralisation and then writing it as كلاسات ‘claassaat’. We annotated these kinds of words as <MAE>, which means a mixed Arabic and English word. Figure 3.5 shows some MAE words found in our corpus.



```
</MAE>البوست</Egypt><MAE>ده لاي حد مش عارف الطريق ازاي ولا عـ
هتكلم علي التوازن في الحياه وازاي تبقي ش
ربنا خلقك في الدنيا ديه لغرض واحد بس وهو ) انك تكو
يعني انتا يالي شايف نفسك ولا حاجه انتـ
</MAE>الموبايل</Egypt><MAE>اللي معاك فيه ملايين</MAE><Egypt>الابليكشنز><MAE>
انتا برضو مليان ملاين الحجات ا
طيب ازاي اعرف ا
هقدر اديهالك في</MAE><Egypt>البوست</Egypt><MAE>ده وهي الاراده وديه تقدر تاخـ
```

Figure 3.5: Some examples of mixed Arabic and English (MAE) found in our corpus.

Detecting Teng and MAE words provided one of the biggest challenges for NLP tools because the issue was unforeseen and we had insufficient training data to provide an effective means to identify these phenomena.

3.2.3 General Corpora

The term ‘*general corpora*’ refers to bodies of text consisting of one language . Because Arabic was the primary language in the current research, we built two Arabic corpora—the SDC and EDC (see Table 3.1)—to be used as general dialect corpora for training language models.

Saudi Dialect Corpus (SDC)

There are many dialectal varieties in Saudi Arabia, such as the Najdi dialect (Arabic: اللهجة النجدية) spoken in the central region of Saudi Arabia by approximately 4 million speakers, the Hejazi dialect (Arabic:اللهجة الحجازية) spoken throughout the Hejaz region (the west region) of the country by around 14 million speakers and the Gulf dialect (Arabic:اللهجة الخليجية) spoken in the east region of Saudi Arabia, which is near the Gulf region, by around 7 million speakers (Simons and Fennig 2017). The Gulf dialect spreads to Bahrain, Qatar, UAE and Iraq (Simons and Fennig 2017). In fact, MSA is used throughout all Saudi regions, being mixed with each dialect. A map of the dialect usage in Saudi Arabia is provided in Figure 3.6.

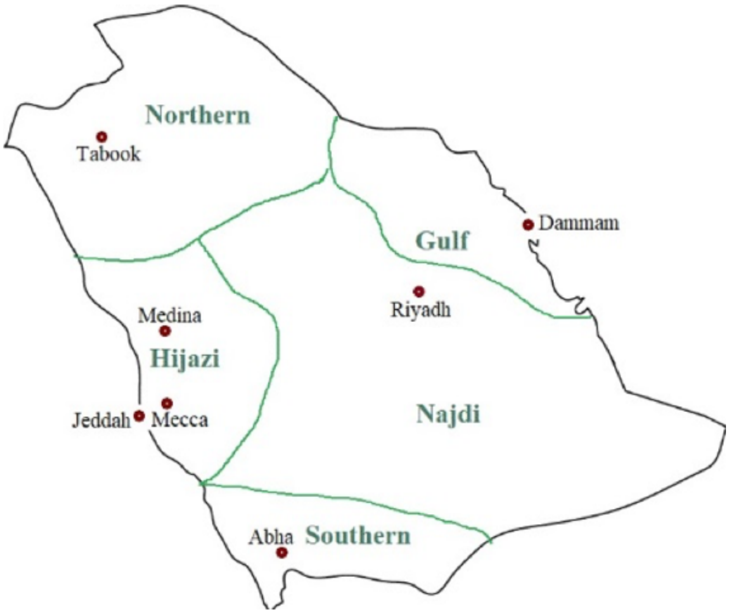


Figure 3.6: A map of Saudi Arabia showing the locations of the dialects (based on Al -Moghrabi (2015).

A 210,396-word corpus called the SDC[2] was built for training the Saudi model,and it contains the mixed dialects of Saudi Arabia. It was collected from social media platforms, such as Facebook and Twitter, and is 2,018 KB in size (see Table 3.1).

Egyptian Dialect Corpus (EDC)

The Egyptian dialect is one of the most widely spoken Arabic dialects. It is used by around 64 million speakers (Simons and Fennig 2017) and as mentioned previous, Egyptian TV and cinema spread their dialect to all Arab countries, so it is considered the most widely understood

[2] https://github.com/TaghreedT/SDC

dialect in the Arab world.

For historical reasons, some frequently used words are shared between the Egyptian dialect and Hejazi dialect (one of the Saudi dialects used in the west of Saudi), which makes distinguishing between these two dialects a challenging task. The EDC[3] that we constructed consists of 218,149 words and is 2,024 KB in size (see Table 3.1). It was also collected from the social media platform Facebook.

3.2.4 Analysing the General Corpora

To analyse the new corpora described above, we investigated the top 10 most frequent words from each corpus. This information allowed us to identify how often the words were used in different corpora and the similarities and differences between them.

Table 3.3 illustrates the top 10 most frequent words from the EDC and SDC. There were found to be some similarities between the Saudi dialect and Egyptian dialect. For example, the word من 'of' is the second most frequent word in both dialects. The word في 'in' is the most frequent word in the Saudi dialect; in contrast, it is the third most frequent word in the Egyptian dialect.

Table 3.3: The top 10 most frequent words from the EDC and SDC.

Rank	SDC		EDC	
	Word	Frequency	Word	Frequency
1	في	4014	و	4508
2	من	3862	من	3443
3	على	3045	في	3122
4	ما	2197	مش	2872
5	بس	2101	اللي	2590
6	انا	1863	ما	1639
7	الي	1650	بس	1553
8	ايش	1345	كل	1456
9	شي	1270	ف	1346
10	والله	1192	عشان	1238

3.3 Classical Arabic Corpus

The second main adjective is to build the NAH corpus. After reviewing the existing Arabic Hadith corpora (in Chapter 2), we found that most concentrate on extracting text from the

[3] `https://github.com/TaghreedT/EDC`

Table 3.4: Hadith corpora comparison.

Corpus	Hadith only	Canonical Books	Non-Canonical Books	Annotated Hadith	Freely available
(Alrabiah et al. 2013)	✕	✓	✓	✕	✓
(Al -Thubaity 2015)	✕	✓	✓	✕	✕
(Alosaimy and Atwell 2017)	✓	✕	✕	✓	✕
(Belinkov, Magidow, Barrón-Cedeño, et al. 2018)	✕	✓	✓	✕	✓
(Mahmood et al. 2018)	✓	✓	✕	✓	✕
(Altammami et al. 2020)	✓	✓	✕	✓	✓
NAH corpus	✓	✕	✓	✓	✓

canonical Hadith books 'الصحاح الستة'. Hence, there is a shortage of research on non-famous Hadith books that have no clear structure. The main motivation for building the NAH corpus was to fill this gap in the research. Table 3.4 compares the existing Hadith corpora and our corpus.

In addition, to evaluate the different classifiers with respect to the automatic classification of the Hadith's authenticity, it was necessary to build an Arabic Hadith corpus that contained samples of authentic Hadith and NAH, which could then be used for training models and testing. We already had access to the LK Hadith corpus (Altammami et al. 2020). Therefore, for a balanced corpus of both positive and negative examples, we needed to collect an Arabic NAH corpus.

Also, as mentioned above, the existing Arabic corpora are quite expensive and/or are of poor quality (Alkahtani 2015). Hence, there was a need to create a free Arabic Hadith corpus.

3.3.1 Creating the Non-authentic Hadith (NAH) Corpus

In the NAH[4] corpus, the goal was to build a corpus containing text from lesser-known Hadith books. These books can be considered challenging for many reasons:

1. They are written in a very old style.
2. They do not have a clear structure.
3. They have not undergone any new revisions, restructuring and editing processes.

Figure 3.7 shows an image of the original 'اللآلئ المصنوعة في الأحاديث الموضوعة' book pages. Figure 3.8 shows a screenshot of the 'Fake Pearls of the Non-Authentic Hadiths' Word file that was used to build our corpus.

[4] https://github.com/TaghreedT/NAH-Corpus

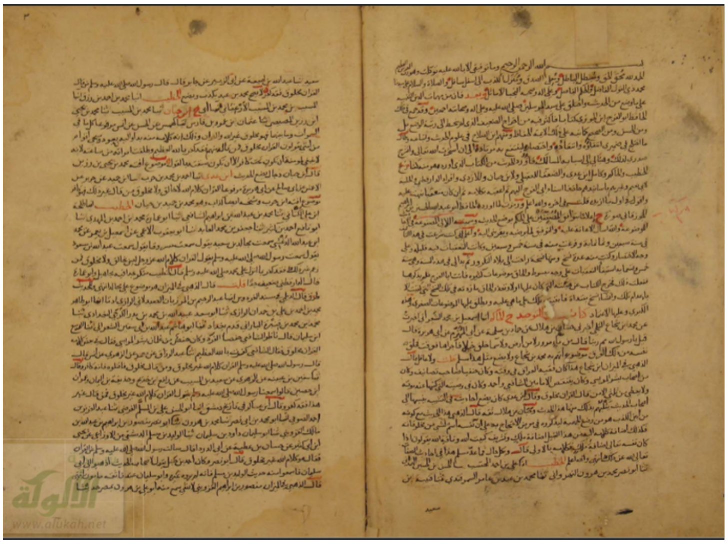
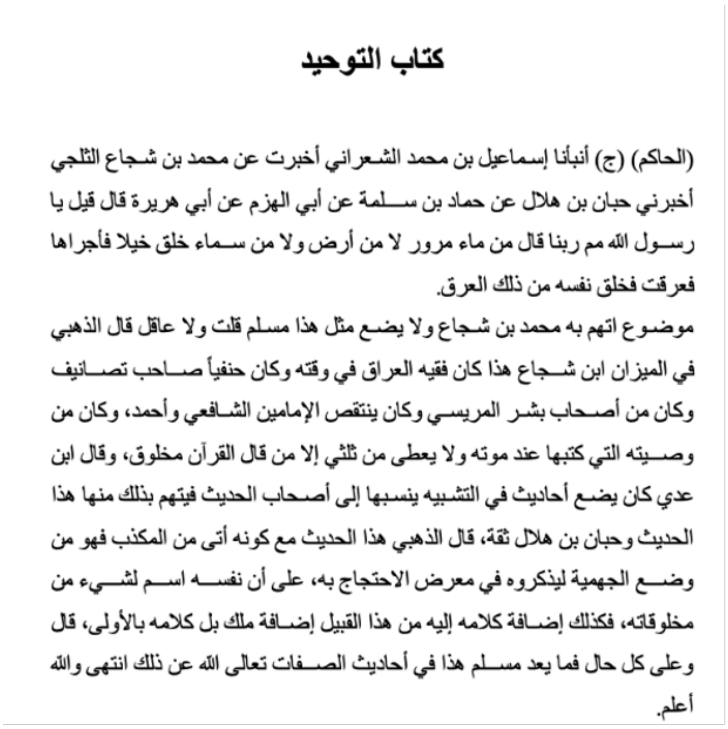


Figure 3.7: An image of the original ‘Fake Pearls of the Non-Authentic Hadiths’ book pages.



كتاب التوحيد

(الحاكم) (ج) أنبأنا إسماعيل بن محمد الشعراني أخبرت عن محمد بن شجاع الثلجي
أخبرني حبان بن هلال عن حماد بن سلمة عن أبي الهزم عن أبي هريرة قال قيل يا
رسول الله مم ربنا قال من ماء مرور لا من أرض ولا من سماء خلق خيلا فأجراها
فعرقت فخلق نفسه من ذلك العرق.

موضوع اتهم به محمد بن شجاع ولا يضع مثل هذا مسلم قلت ولا عاقل قال الذهبي
في الميزان ابن شجاع هذا كان فقيه العراق في وقته وكان حنفياً صاحب تصانيف
وكان من أصحاب بشر المريسي وكان ينتقص الإمامين الشافعي وأحمد، وكان من
وصيته التي كتبها عند موته ولا يعطى من ثلثي إلا من قال القرآن مخلوق، وقال ابن
عدي كان يضع أحاديث في التشبيه ينسبها إلى أصحاب الحديث فيتهم بذلك منها هذا
الحديث وحبان بن هلال ثقة، قال الذهبي هذا الحديث مع كونه أتى من المكذب فهو من
وضع الجهمية ليذكروه في معرض الاحتجاج به، على أن نفسه اسم لشيء من
مخلوقاته، فكذلك إضافة كلامه إليه من هذا القبيل إضافة ملك بل كلامه بالأولى، قال
وعلى كل حال فما يعد مسلم هذا في أحاديث الصفات تعالى الله عن ذلك انتهى والله
أعلم.

Figure 3.8: A screenshot of the ‘Fake Pearls of the Non-Authentic Hadiths’ Word file that was used to build our corpus.

Our corpus is called a NAH corpus because of the large number of NAH compared with authentic Hadith. Most Hadith studies have focused on the six famous Hadith books that have a clear structure. Hence, the main feature of the NAH corpus is that it contains 1,621,423 words from 15 non-famous Hadith books that have no clear structure. Over 4,000 Hadiths were annotated manually according to the Hadiths’ Isnads and Matans, in addition to author comments, Hadith rank, Hadith authenticity and Hadith topic. These Hadiths were divided into over 7,000 Hadith records because some Hadiths were classified as Hadith blocks (discussed in Section 3.3.3).

A Python application was utilised to automatically extract *N1*, the first book in this corpus, from the *islamweb.net* website. After we had finished annotating this book, we found out that all the Hadith books containing authentic and non-authentic Hadiths had been removed from the *islamweb.net* website for an audit process. Therefore, we moved to the *almeshkat.net* website. Several books were downloaded as Word files and converted to comma-separated value (CSV) files (see Table 3.5). Table 3.5 shows the corpus contents. Here, *No.* refers to the book number in this corpus, followed by *Book ID*. In the *Book ID* field, *N* refers to a non-authentic word, while _1 and _2 means that the book has two parts (_1 is part one and _2 is part two). This is followed by *Book title* and *Author*. Some books have an Isnad, Matan and comments, while others just have a Matan and comments, so we added *Book contents* to clarify these contents. Then, we added *Hadith's type*, which clarifies whether the book has an authentic Hadith and NAH or just NAH. Then, *No. of words*, *Annotated* or not and *verified* by two annotators or not is given,followed at last by *Website*.

Table 3.5: The NAH corpus contents.

No.	Book ID	Book Title	Author	Book Contents	Hadith's Type	No. of words	Annotated	verified	website
1	N1	الأباطيل والمناكير والصحاح والمشاهير	أبو عبد الله الهمذاني الجورقاني	Isnad/Matan/Comments	authentic and NAH	121,080	Yes	Yes	islamweb.net
2	N2	مائة حديث ضعيف وموضوع منتشرة بين الخطباءوالوعاظ	إحسان العتيبي	Matan/Comments	NAH	2,898	Yes	No	almeshkat.net
3	N3_1	اللآلئ المصنوعة في الأحاديث الموضوعة الجزء الأول ـ ط دار المعرفة	جلال الدين السيوطي	Isnad/Matan/Comments	authentic and NAH	15,421	Yes	Yes	almeshkat.net
4	N3_2	اللآلئ المصنوعة في الأحاديث الموضوعة الجزء الثاني ـ ط دار المعرفة	جلال الدين السيوطي	Isnad/Matan/Comments	authentic and NAH	151,382	Yes	Yes	almeshkat.net
5	N4	الأحاديث الضعيفة في كتاب رياض الصالحين	إحسان العتيبي	Isnad/Matan/Comments	NAH	5,675	Yes	No	almeshkat.net
6	N5	-الجد الحثيث في بيان ما ليس بحديث ـ ت: أبو زيد دار الراية	أحمد بن عبد الكريم العامري	Matan/Comments	NAH	16,382	Yes	No	almeshkat.net
7	N6	الفوائد المجموعة في الأحاديث الموضوعة ـ ط العلمية	الإمام محمد بن علي الشوكاني	Matan/Comments	NAH	139,786	Yes	Yes	almeshkat.net
8	N7	معرفة التذكرة في الأحاديث الموضوعة ـ مؤسسة الكتب الثقافية	لابن طاهر المقدسي	Matan/Comments	NAH	115,672	No	No	almeshkat.net
9	N8	جامع الأحاديث القدسية (الضعيفة) دار الريان للتراث	عصام الدين الصابطي	Matan/Comments	NAH	246,141	No	No	almeshkat.net
10	N9	ضعيف سنن الترمذي	محمد ناصر الألباني	Isnad/Matan/Comments	NAH	663,783	No	No	almeshkat.net
11	N10	الموضوعات ـ دار المأمون للتراث دمشق	الحسن بن محمد الصغاني	Matan/Comments	NAH	13,508	No	No	almeshkat.net
12	N11	النخبة البهية في الأحاديث المكذوبة على خير البرية ـ المكتب الإسلامي	محمد الأمير الكبير المالكي	Matan/Comments	NAH	13,508	No	No	almeshkat.net
13	N12	المصنوع في معرفة الحديث الموضوع	علي القاري الهروي المكي	Matan/Comments	NAH	33,037	No	No	almeshkat.net
14	N13	أحاديث الإحياء التي لا أصل لها	الإمام تاج الدين السبكي	Matan	NAH	55,917	No	No	almeshkat.net
15	N14	اللؤلؤ الموصوع فيما لا أصل له أو بأصله موضوع ط دار البشائر + ط قديمة	الشيخ أبي المحاسن القاوقجي	Matan/Comments	NAH	27,233	No	No	almeshkat.net
	Total					1,621,423			

Figure 3.9 shows the most frequently used words among all Hadith books in the NAH corpus.

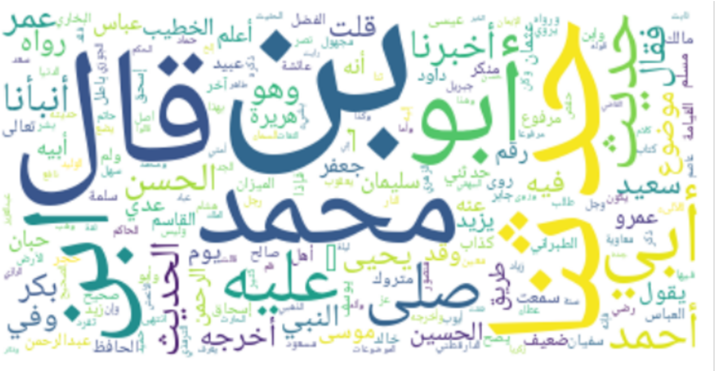


Figure 3.9: The NAH corpus word cloud.

3.3.2 Methodology

The web as corpus method (Kilgarriff and Grefenstette 2003) was chosen to collect Hadiths from the *islamweb.net* and *almeshkat.net* websites. Because the web texts are free and written by a wide variety of writers, there is a lack of interest in proofreading them (Kilgarriff and Grefenstette 2003). We found different mistakes in our corpus, such as missing spaces رضيالله، عنها، عنمسروق , missing letters الل and spelling errors مرضوع، اتهى، مفوعاً . After comparing the *N3_1* Word file with the original book pdf file, we found some Hadiths were missing in the Word file. We left these errors as they were written in the source.Figure 3.10 shows an example of missing space in *N5*.

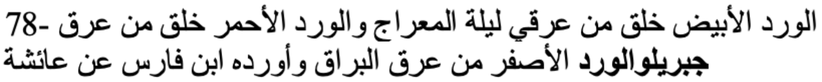

الورد الأبيض خلق من عرقي ليلة المعراج والورد الأحمر خلق من عرق -78
جبريلوالورد الأصفر من عرق البراق وأورده ابن فارس عن عائشة

Figure 3.10: An example of missing space in *N5* (in bold).

3.3.3 Corpus Annotation

The NAH corpus contains two primary folders. The annotated folder contains seven CSV files encompassing the Hadith books that have been manually annotated. The unannotated folder contains five CSV files that contain the Hadith books that have not been annotated (see Figure 3.11).

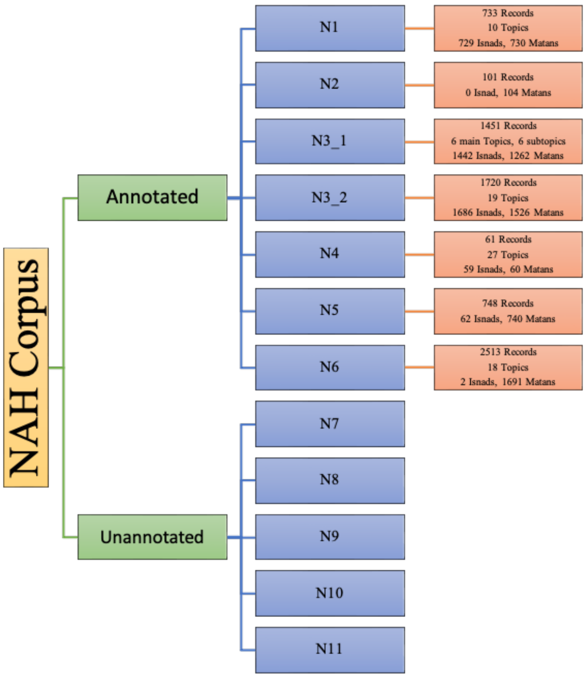
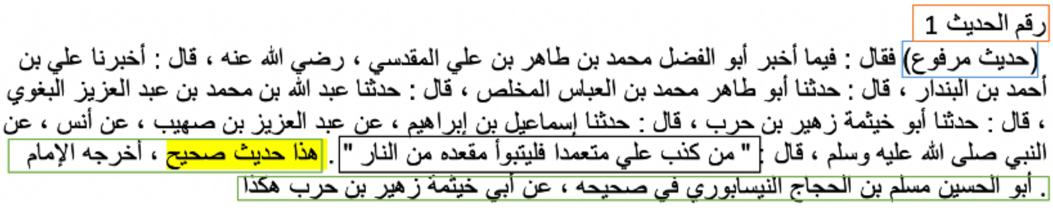


Figure 3.11: The NAH corpus structure.

Every Hadith in the first folder has eight primary features or attributes: *No.*, *Full Hadith*, the *Isnad*, the *Matan*, the *Authors comments*, the *Hadith rank*, *Authenticity* (the annotator copied the Hadith authenticity from Hadith book which was acknowledged by Hadith scholars) and *Topic*. A description of the NAH corpus features is provided in Table 3.6. Figure 3.12 shows how the author wrote each Hadith; the Hadith number (red square) is followed by the Hadith rank (blue square). After that, we show that they wrote the Isnad (between the blue square and the black square), followed by the Matan, which was written between parentheses (black square). Finally, at the end of Hadith, the author describes the authenticity of each Hadith (green square). Table 3.7 shows an example of an annotated Hadith extracted from *N3_1*.



رقم الحديث 1
(حديث مرفوع) فقال : فيما أخبر أبو الفضل محمد بن طاهر بن علي المقدسي ، رضي الله عنه ، قال : أخبرنا علي بن
أحمد بن البندار ، قال : حدثنا أبو طاهر محمد بن العباس المخلص ، قال : حدثنا عبد الله بن محمد بن عبد العزيز البغوي
، قال : حدثنا أبو خيثمة زهير بن حرب ، قال : حدثنا إسماعيل بن إبراهيم ، عن عبد العزيز بن صهيب ، عن أنس ، عن
النبي صلى الله عليه وسلم ، قال : " من كذب علي متعمدا فليتبوأ مقعده من النار " . هذا حديث صحيح ، أخرجه الإمام
. أبو الحسين مسلم بن الحجاج النيسابوري في صحيحه ، عن أبي خيثمة زهير بن حرب هكذا

Figure 3.12: A screenshot of the first Hadith from *N1*.

In some Hadith books, the Hadiths have been classified by their topics, so we added the *Topic* feature. In the first book (*N1*), we noticed that there was confusion in this classification because

Table 3.6: Features of the NAH corpus.

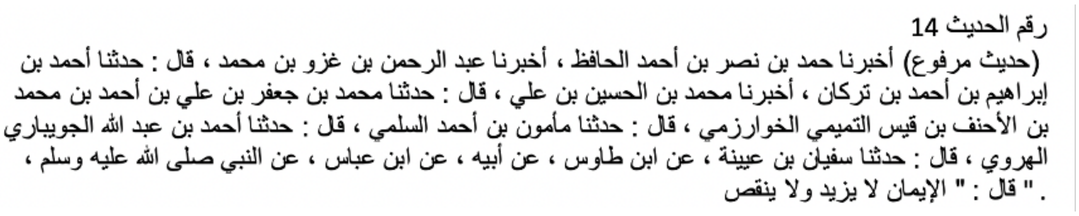
Features	Description
No.	The Hadith reference number.
Full Hadith	The Hadith as it appears in the book without annotations
Isnad	The chain of narrators
Matan	The act of the Prophet Muhammad
Authors Comments	The author describes the authenticity of each Hadith
Hadith Rank	The Hadith Rank (Maqtuمقطوع , Mawquf موقوف and Marfo مرفوع) or Hadith degree (ضعيف, موضوع, صحيح and so on)
Authenticity	Whether this Hadith is authentic or non-authentic
Topic	The chapter title

Table 3.7: An example of an annotated Hadith extracted from *N3_1*.

No.	Full Hadith	Isnad	Matan	Authors Comments	Degree	Authenticity	Topic
1	(الحاكم) (ج) أنبأنا إسماعيل بن محمد الشعراني أخبرت عن محمد بن شجاع الثلجي أخبرني حبان بن هلال عن حماد بن سلمة عن أبي الهزم عن أبي هريرة قال قيل يا رسول الله م ربنا قال من ماء مرور لا من أرض ولا من سماء خلق خيلا فأجراها	أنبأنا إسماعيل بن محمد الشعراني أخبرت عن محمد بن شجاع الثلجي أخبرني حبان بن هلال عن حماد بن سلمة عن أبي الهزم عن أبي هريرة قال	قيل يا رسول الله م ربنا قال من ماء مرور لا من أرض ولا من سماء خلق خيلا فأجراه عرقت فخلق نفسه	موضوع اتهم به محمد بن شجاع ولا يضع مثل هذا مسلم قلت ولا عاقل قال الذهبي في الميزان ابن شجاع هذا كان فقيه العراق في وقته وكان حنفياً صاحب تصانيف وكان من أصحاب بشر المريسي وكان ينتقص الإمامين الشافعي وأحمد، وكان من وصيته التي ك	موضوع	non-authentic	كتاب التوحيد

in the prayer ‘الصلاة’ topic, there are some Hadiths about the charity ‘الزكاة’. In addition, in the fasting ‘الصوم’ topic, there are some Hadiths about the pilgrimage ‘الحج’. This may cause a problem if we try to use automatic topic classifications.

In Hadith books, there are different types of Hadiths, such as *Maqtu* ‘مقطوع’, *Mawquf* ‘موقوف’ and *Marfo* مرفوع’. The *Maqtu* Hadiths refer to sayings, actions and explanations attributed to a man who met the Prophet Muhammad’s friends (a successor) (Ibn Al-Salah 1236). The *Mawquf* Hadiths describe a statement or action of the Prophet Muhammad’s friends (a sahaba). The *Marfo* Hadiths refer to any action, saying or order that was done by Prophet Muhammad and has been delivered through a chain of narrators (Ibn Al-Salah 1236). All these types of Hadiths could be either authentic or not. To make this determination, Hadith scholars follow certain rules (discussed in Section 2.2.2). As previously stated, the author describes the authenticity of each Hadith, but some Hadiths lack comments. Therefore, we cannot know their authenticity. For example, in Figure 3.12, the beginning of the Hadith, in the *N1*, lists *Marfo*, and at the end, the author states that this is an authentic Hadith (highlighted in yellow). By contrast, in Figure 3.13, the author does not describe the authenticity of this Hadith.



رقم الحديث 14
(حديث مرفوع) أخبرنا حمد بن نصر بن أحمد الحافظ ، أخبرنا عبد الرحمن بن غزو بن محمد ، قال : حدثنا أحمد بن
إبراهيم بن أحمد بن تركان ، أخبرنا محمد بن الحسين بن علي ، قال : حدثنا محمد بن جعفر بن علي بن أحمد بن محمد
بن الأحنف بن قيس التميمي الخوارزمي ، قال : حدثنا مأمون بن أحمد السلمي ، قال : حدثنا أحمد بن عبد الله الجويباري
الهروي ، قال : حدثنا سفيان بن عيينة ، عن ابن طاوس ، عن أبيه ، عن ابن عباس ، عن النبي صلى الله عليه وسلم ،
قال : " الإيمان لا يزيد ولا ينقص " .

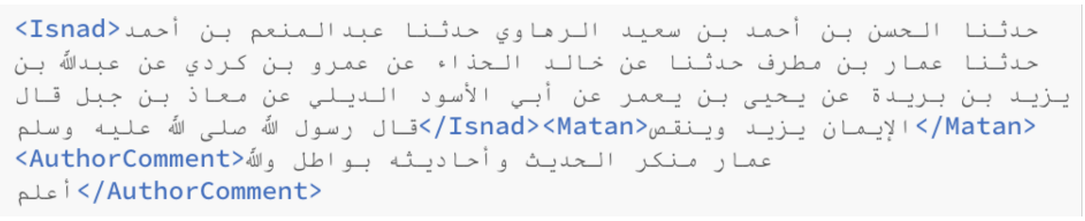
Figure 3.13: An example of a Hadith with the author’s comment missing.

Table 3.8 shows the expressions that were used by the authors to describe the authenticity of each Hadith.

Table 3.8: Expressions used by the authors to describe the authenticity of each Hadith.

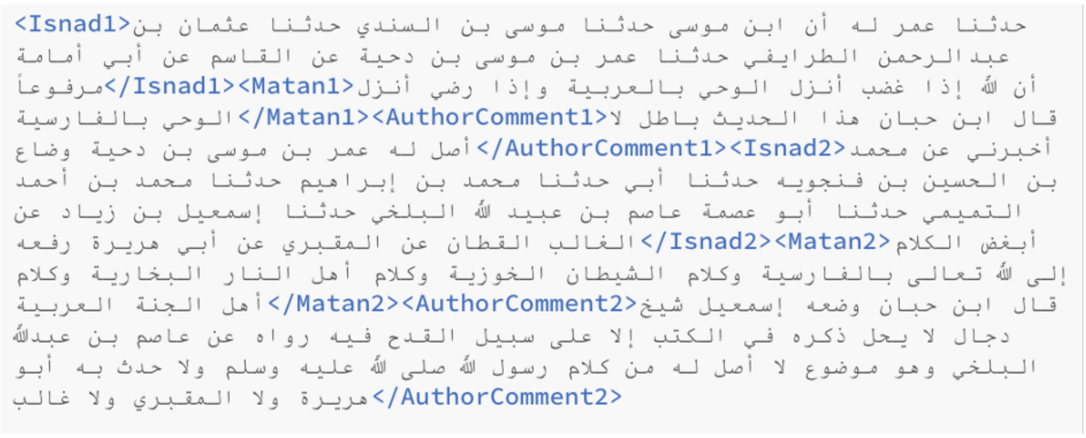
Authentic	Non-authentic
صحيح	موضوع
صحيح حسن	غير صحيح
حسن	باطل
محتمل التحسين	مضطرب
قوي	منكر
جيد	كذب
رواته ثقات	ضعيف
	ليس لهذا الحديث أصل
	موقوف منكر

In the annotating process, two kinds of Hadith have been found in our corpus; *simple Hadith* and *Hadith block*. The *simple Hadith* refers to a Hadith that has an Isnad, a Matan and author's comment (if found). Figure 3.14 shows an example of a simple Hadith. The *Hadith block* refers to a complex kind of Hadith that contains several Isnads, Matans or author comments that were written sequentially Figure 3.15illustrates an example of a Hadith block. The *simple Hadith* has been represented as a one Hadith record in the NAH corpus (see Table 3.7), while the *Hadith block* has been represented as a several Hadith records, here depending on how many Isnads, Matans or author comments there are (see Table 3.9). In Table 3.9, the Hadith block was separated into three records because it has three Isnads and three Matans, while the full Hadith was written just in the first record (7_1). The longest Hadith block, was found in *N*3_1, is represented in 18 Hadith records.



```
<Isnad>حدثنا الحسن بن أحمد بن سعيد الرهاوي حدثنا عبدالمنعم بن أحمد
حدثنا عمار بن مطرف حدثنا عن خالد الحذاء عن عمرو بن كردي عن عبدالله بن
يزيد بن بريدة عن يحيى بن يعمر عن أبي الأسود الديلي عن معاذ بن جبل قال
قال رسول الله صلى الله عليه وسلم</Isnad><Matan>الإيمان يزيد وينقص</Matan>
<AuthorComment>عمار منكر الحديث وأحاديثه بواطل والله
أعلم</AuthorComment>
```

Figure 3.14: An example of a simple Hadith extracted from *N3_1*.



```
حدثنا عمر له أن ابن موسى حدثنا موسى بن السندي حدثنا عثمان بن<Isnad1>
عبدالرحمن الطرايفي حدثنا عمر بن موسى بن دحية عن القاسم عن أبي أمامة
أن الله إذا غضب أنزل الوحي بالعربية وإذا رضي أنزل</Isnad1><Matan1>مرفوعاً
قال ابن حبان هذا الحديث باطل لا</Matan1><AuthorComment1>الوحي بالفارسية
أخبرني عن محمد</AuthorComment1><Isnad2>أصل له عمر بن موسى بن دحية وضاع
بن الحسين بن فنجويه حدثنا أبي حدثنا محمد بن إبراهيم حدثنا محمد بن أحمد
التميمي حدثنا أبو عصمة عاصم بن عبيد الله البلخي حدثنا إسمعيل بن زياد عن
أبغض الكلام</Isnad2><Matan2>الغالب القطان عن المقبري عن أبي هريرة رفعه
إلى الله تعالى بالفارسية وكلام الشيطان الخوزية وكلام أهل النار البخارية وكلام
قال ابن حبان وضعه إسمعيل شيخ</Matan2><AuthorComment2>أهل الجنة العربية
دجال لا يحل ذكره في الكتب إلا على سبيل القدح فيه رواه عن عاصم بن عبدالله
البلخي وهو موضوع لا أصل له من كلام رسول الله صلى الله عليه وسلم ولا حدث به أبو
هريرة</AuthorComment2>ولا المقبري ولا غالب
```

Figure 3.15: An example of a Hadith block extracted from *N3_1*.

Table 3.9: An example of how the Hadith block was annotated in *N3_1*.

No.	Full Hadith	Isnad	Matan	Authors Comments	Degree	Authenticity	Topic
7_1	حدثنا عبدالرزاق عن معمر عن الزهري عن أنس قال قال رسول الله صلى الله عليه وسلم القرآن كلام الله غير خالق ولا مخلوق ومن قال مخلوق فاقتلوه فإنه كافر، وقال حدثنا سفيان بن عيينة عن الزهري عن سعيد بن المسيب عن رافع بن خديج وحذيفة بن اليمان وعمران بن حصين قالوا سمعنا رسول الله صلى الله عليه وسلم يقول...	حدثنا عبدالرزاق عن معمر عن الزهري عن أنس قال قال رسول الله صلى الله عليه وسلم	القرآن كلام الله غير خالق ولا مخلوق ومن قال مخلوق فاقتلوه فإنه كاف			non-authentic	كتاب التوحيد
7_2		وقال حدثنا سفيان بن عيينة عن الزهري عن سعيد بن المسيب عن رافع بن خديج وحذيفة بن اليمان وعمران بن حصين قالوا سمعنا رسول الله صلى الله عليه وسلم يقول	لقرآن كلام الله غير خالق ولا مخلوق فمن قال غير هذا فقد كفر			non-authentic	كتاب التوحيد
7_3		وقال ابن عساكر في تاريخ دمشق أنبأنا أبو الحسن علي بن السلم الفرضي حدثنا عبدالعزيز أحمد لصوفي أنبأنا أبو محمد بن أبي نصر حدثنا محمد بن هرون حدثنا أبو نصر منصور بن إبراهيم بن عبدالله بن مالك القزويني	هو كلام الله غير مخلوق	قال أبو نصر وكان أحمد بن حنبل يقول لأصحاب الحديث اذهبوا إلى أبي سليمان عندنا ثقة مأمون، انتهى قال الذهبي في الميزان منصور بن إبراهيم القزويني لا شئ سمع منه أبو علي بن هارون بمصر حديثاً باطلاً، قال الحافظ بن حجر في لسانه هو هذا الحديث، انتهى. وقد وجدت له متابعاً قال		non-authentic	كتاب التوحيد

3.3.4 Corpus Evaluation

This section describes various experimental analyses conducted to evaluate the corpus. First, a cross-corpus evaluation was used to compare the classification results of the NAH corpus with other Hadith corpora using different CML and DL classifiers. This assisted in verifying Hadith components (Isnad and Matan) by comparing them against existing Hadith corpora. Second, to verify the quality of the annotating, we applied an inter-annotator agreements (IAA) analysis.

Cross-corpus Evaluation

To evaluate the NAH corpus, we compared it with another existing corpus with similar features: the LK corpus Altammami et al. (2020). In this experiment, we used one corpus as a training set and the other corpus as a testing set. The experiment used NB, DT (J48) algorithms using the Weka toolkit (Hall et al. 2009), CNN and LSTM classifiers.

Table 3.10 shows that the NAH corpus identified 98% and 92% of the LK corpus using NB and J48, respectively. Whereas the LK corpus identified 87% and 82% of the NAH corpus using NB and J48, respectively. Also, the NAH corpus identified 98% and 99% of the LK corpus using CNN and LSTM, respectively. The LK corpus identified 90% and 98% of the NAH corpus using CNN and LSTM, respectively. This demonstrates that even when using different classifiers, training models with the NAH corpus results in higher accuracy rates than training with the LK corpus.

Table 3.10: Cross-corpus evaluation using NB, J48, CNN and LSTM trained on the training datasets (rows) and tested on testing datasets (columns).

Classifier	Dataset	NAH	LK
NB	NAH	-	98.60%
	LK	87.42%	-
J48	NAH	-	92.80%
	LK	82.74%	-
CNN	NAH	-	98.39%
	LK	90.54%	-
LSTM	NAH	-	99.57%
	LK	98.06%	-

Inter-annotator Agreements

Annotation of the NAH corpus was carried out by two annotators with Arabic and Islamic backgrounds. To validate the quality of their annotation, the Kappa coefficient, k (Cohen 1960) was chosen to calculate the IAA between the two annotators, as follows:

$$k = \frac{P_o - P_e}{1 - P_e} \tag{3.1}$$

where P_o is the probability of the actual agreement between annotators, and P_e the probability of the expected random agreement.

$$P_e = \frac{1}{N^2} \sum_q n_{A1q} \times n_{A2q} \tag{3.2}$$

where N is the total number of annotated Hadith records and n_{Aiq} is the number of Hadith records which annotator A_i labelled with tag q.

The process of using annotators was quite expensive, so we provided only three datasets to the them: $N3\_1$, $N3\_2$ and $N6$ from the NAH. Then, the Kappa coefficient was calculated for a total of 4,338 Hadith records and obtained Kappa values between 0.9842 and 0.9983, as shown in Table 3.11, which indicates perfect agreement, according to Landis and Koch (1977).

The reason for the disagreement in some complex Hadith records is the confusion between the Matan and author's comment, such as به، فذكر نحوه، فذكر الحديث and so on. To solve this, the annotators reached an agreement that the accurate label for them should be the author's comment; then, the annotation label was modified to reflect the accurate label.

Our experiments in the following chapters will use the NAH corpus produced after solving these disagreements.

Table 3.11: Inter-annotator agreement, disagreement and Kappa coefficient value for the NAH corpus.

File Name	Agreement (records)	Disagreement (records)	Kappa
N3_1	1434	23	0.9842
N3_2	1714	6	0.9965
N6	1159	2	0.9983

3.3.5 Analysing the NAH corpus

To analyse the new corpus described above, we investigated the top 10 most frequently used words from each book in the annotated folder. This information allowed us to identify how often words have been used in different books and the similarities and differences between them. Table 3.12 illustrates the top 10 most frequently used words from each book in the NAH. There are some similarities between them. For example, the word الله (Allah) is a common frequently used word in all NAH corpus books, and it is rank between the first and fifth most frequently used word. Also, the words قال (said) and بن (son) are common frequently used words in these books. The most frequently used words from $N2$ are different than other books because this book focused on referring each non-authentic Matan to its original book.

Table 3.12: The top 10 most frequently used words from each book in the NAH.

Dataset	Most frq. Words			Dataset	Most frq. Words		
N1	1	بن	9196	N2	1	الضعيفة	53
	2	قال	4324		2	الله	46
	3	الله	3264		3	ضعيف	45
	4	حدثنا	2641		4	الموضوعات	41
	5	محمد	2232		5	موضوع	36
	6	أخبرنا	2046		6	قال	32
	7	عبد	1992		7	لابن	29
	8	أبو	1731		8	أصل	20
	9	حديث	1359		9	تذكرة	19
	10	أبي	1339		10	الفوائد	16
$N3_1$	1	بن	10557	$N3_2$	1	بن	10090
	2	حدثنا	4924		2	حدثنا	5191
	3	اللّه	3227		3	اللّه	3335
	4	قال	3143		4	قال	3129
	5	محمد	2477		5	محمد	2423
	6	أبو	1965		6	ابن	2368
	7	ابن	1794		7	أبو	1936
	8	أبي	1738		8	أبي	1682
	9	عليه	1345		9	عليه	1395
	10	علي	1252		10	صلى	1171
N4	1	الله	242	N5	1	الله	275
	2	قال	109		2	قال	275
	3	عليه	84		3	الجد	271
	4	صلى	78		4	الحثيث	231
	5	وسلم	77		5	ابن	153
	6	بن	73		6	بحديث	138
	7	ضعيف	71		7	كلام	129
	8	رضي	58		8	بن	95
	9	رواه	56		9	ولا	76
	10	رسول	56		10	حديث	71
N6	1	بن	2975				
	2	ابن	2178				
	3	حديث	1781				
	4	الله	1774				
	5	قال	1721				
	6	أبي	968				
	7	رواه	867				
	8	وفي	768				
	9	وقال	754				
	10	وهو	721				

3.4 Challenges

Of note, while building our corpora, we encountered several challenges. The first challenge pertained to the SDC. Saudi users only recently started using colloquial Arabic on social networks. Thus, there were not enough Facebook data to build the SDC. This led us to add Twitter data because it is the second most popular social network platform in Saudi Arabia. A second challenge was building the Saudi and Egyptian dialect corpora. Because Arabic posts contain non-Arabic words, such as MAE and TEng (discussed in Section 3.2.2), hashtags, URLs and emojis, we spent a lot of time performing extensive text cleaning. A third challenge pertained to producing the annotated BAEC. This was a complex task because Arabic dialects do not have clear spelling standards and conventions and have a lot of emojis, URLs, English numbers and non-Arabic words. The fourth challenge pertained to producing the NAH. The *islamweb.net* website was chosen to build our corpus because it has several Hadith books that contain authentic and non-authentic Hadiths. After we automatically extracted the first book ($N1$) and finished annotating it, we found out that all Hadith books containing authentic and non-authentic Hadiths have been removed from the website for an audit process. Therefore, we moved to the *almeshkat.net* website. Annotating the lesser-known Hadith books can be considered challenging because they have been written in a very old style; do not have a clear structure; and have not undergone new revision, restructuring and editing processes. Finally, annotating the NAH corpus was quite expensive, so we provided only three books to the annotators who took part in this research.

3.5 Conclusion

This chapter described the production of several new Arabic corpora, which required substantial time and effort. These new corpora represent valuable and rich resources for the Arabic NLP community. This chapter first discussed the production of dialectal Arabic corpora, as follows: the BAEC consists of 45,251 words, SDC 210,396 words and EDC 218,149 words. The main motivation for building these corpora was to evaluate a compression-based approach and Classical ML classifiers for the automatic detection of code-switching, We analysed these corpora by investigating the top 10 most frequently used words from each corpus. We found some similarities and differences between them.

This chapter discussed the production of the NAH corpus in detail. The NAH contains Arabic

Hadith from lesser-known Hadith books, providing a new resource for the Hadith research community. The main motivation for building this corpus was to evaluate a compression-based approach and CML and DL classifiers for the automatic classification of Arabic Hadiths. The NAH consists of 1,621,423 words from 15 non-famous Hadith books. Six books went through a paid annotation process. Each book took approximately one month (80 hours) to complete. Then, we discussed several challenges that we encountered while building our corpora.

In the following chapter, we utilise these new Arabic corpora to evaluate a compression-based approach and a CML approach for the automatic detection of code-switching in Arabic text.

Chapter 4

Detect Code-Switching in Varieties and Dialects

Code-switching in written natural language text occurs when the author chooses to switch from one language to at least one other. The occurrence of code-switching in online communication, when a writer switches among multiple languages, presents a challenge for natural language processing (NLP) tools, since they are designed for texts written in a single language. To answer the challenge, this chapter presents detailed research on ways to detect code-switching in Arabic text automatically.

This chapter is an extension of the papers published as follows:

1. *Tarmom T, Teahan W, Atwell E, Alsalka MA. 2020. Compression versus traditional machine learning classifiers to detect code-switching in varieties and dialects: Arabic as a case study. Natural Language Engineering. 26(6), pp. 663-676*

2. *Tarmom T; Teahan W; Atwell E; Alsalka M (2019) Code-Switching in Arabic Dialect Corpora: Compression vs Traditional Machine Learning Classifiers to Detect Code-switching. The International Corpus Linguistics Conference 2019*

The work in this chapter uses the Tawa toolkit (Teahan 2018) (Teahan, 2018), which uses the Prediction by Partial Matching (PPM) compression scheme; it also uses the Waikato Environment for Knowledge Analysis (Weka) data analytic tool as a second method for the automatic detection of code-switching in Arabic text. It provides a comparison between the classical machine learning classifiers algorithm which is the Support Vector Machine (SVM) which is called Sequential Minimal Optimization (SMO) in Weka and the PPM compression-based approach. This chapter outlines the experiments performed on Arabic Facebook text to evaluate the PPM classifier produced by the Tawa toolkit and the SVM classifier provided by Weka. Also, it provides a conclusion for this study.

4.1 Experimental Setup

Three experiments were performed to evaluate the compression-based approach (provided by Tawa) and of classical ML classifiers such as the SVM classifier (provided by Weka) to detect code-switching in Arabic Facebook text: (1) detect code-switching between the Egyptian dialect and English; (2) detect code-switching among the Egyptian dialect, the Saudi dialect and English; and (3) detect code-switching among the Egyptian dialect, the Saudi dialect, MSA and English.

Choosing suitable training corpora is the first step in building language models. As indicated in section 2.4.1, the SDC and EDC were created for this purpose. The corpus selected for training MSA was built by Alkahtani (2015). The Brown corpus was selected for training English.

The BAEC corpus (described in section 3.2.2) was used as a testing corpus. First, a cleaning process removed all MAE and TEng words (described in section 3.2.2) because we did not have enough training data to provide an effective means for identifying these phenomena. Numbers, emojis and punctuation marks were also removed, leaving only pure Arabic and English to process because we thought this would reduce the errors and enhance the accuracy. A confusion matrix was used to evaluate the performance of the automatic detection of code-switching.

4.2 Experiments and Results

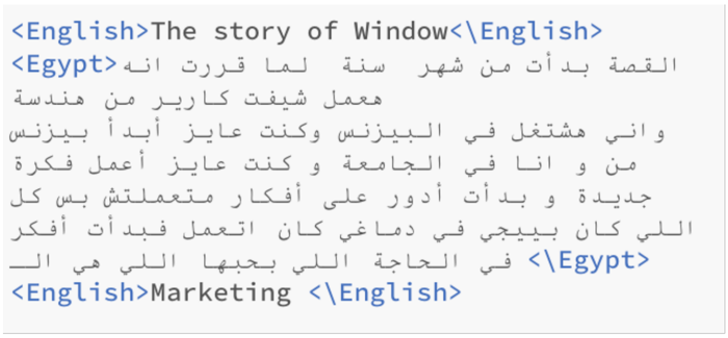
4.2.1 Detecting Code-switching between the Egyptian dialect and English

The first experiment was conducted to evaluate the PPM compression algorithm and focused on the detection of code-switching between the Egyptian dialect and English. The testing text was manually extracted from the BAEC corpus for cases containing only the Egyptian dialect and English, around 17,761 words and 79,975 characters.

The automatic detection of code-switching between the Egyptian dialect and English, here using the PPM compression algorithm, obtained an accuracy of 99.8%, an average recall of 99.6%, an average precision of 99.9% and an average F-measure of 99.8%. Table 4.1 shows the results of this experiment. A sample output from the PPM classifier is shown in Figure 4.1.

Table 4.1: The results of the first experiment.

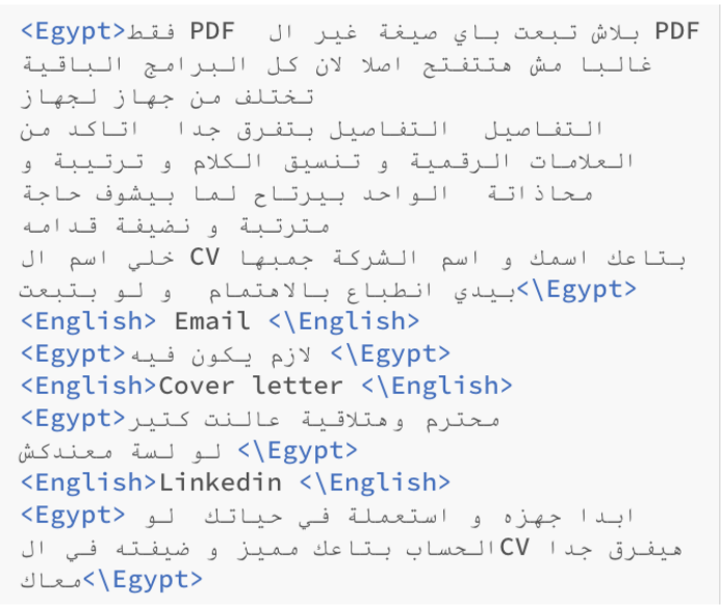
	Accuracy	Precision	Recall	F-Measure
PPM	**0.998**	**0.999**	**0.996**	**0.998**
SVM (WordTokenizer)	0.550	0.671	0.550	0.440
SVM (UniGram)	0.975	0.975	0.975	0.975
SVM (BiGram)	0.827	0.851	0.827	0.822
SVM (TriGram)	0.645	0.687	0.645	0.607
SVM (FourGram)	0.558	0.603	0.558	0.475



```
<English>The story of Window<\English>
<Egypt>القصة بدأت من شهر  سنة  لما قررت انه
هعمل شيفت كارير من هندسة
واني هشتغل في البيزنس وكنت عايز أبدأ بيزنس
من و انا في الجامعة و كنت عايز أعمل فكرة
جديدة و بدأت أدور على أفكار متعملتش بس كل
اللي كان بييجي في دماغي كان اتعمل فبدأت أفكر
في الحاجة اللي بحبها اللي هي الـ <\Egypt>
<English>Marketing <\English>
```

Figure 4.1: Sample output from the first experiment's output using PPM.

However, we noticed that most occurrences of English text that were incorrectly predicted as an Egyptian dialect were abbreviations, such as *CV*, *PDF* and *BBC*, because the Brown corpus, which has been built from American English written texts, does not include abbreviations. The use of a new English corpus for abbreviations should further reduce the number of errors. Figure 4.2 shows that the text *PDF* and *CV* were predicted as an Egyptian dialect.



```
<Egypt>PDF بلاش تبعت باي صيغة غير ال PDF فقط
غالبا مش هتتفتح اصلا لان كل البرامج الباقية
تختلف من جهاز لجهاز
التفاصيل  التفاصيل بتفرق جدا  اتاكد من
العلامات الرقمية و تنسيق الكلام و ترتيبة و
محاذاتة  الواحد بيرتاح لما بيشوف حاجة
مترتبة و نضيفة قدامه
بتاعك اسمك و اسم الشركة جمبها CV خلي اسم ال
<\Egypt>بيدي انطباع بالاهتمام و لو بتبعت
<English> Email <\English>
<Egypt>لازم يكون فيه <\Egypt>
<English>Cover letter <\English>
<Egypt>محترم وهتلاقية عالنت كتير
لو لسة معندكش <\Egypt>
<English>Linkedin <\English>
<Egypt> ابدا جهزه و استعملة في حياتك لو
هيفرق جدا CVالحساب بتاعك مميز و ضيفته في ال
معاك<\Egypt>
```

Figure 4.2: Sample errors from the first experiment's output.

Alshutayri, Atwell, et al. (2016) pointed out that the SVM algorithm achieved the best accuracy rate when classifying Arabic dialects, so we used this classifier in our experiments. Hence, we repeated this experiment using Weka for the best results. We used the SVM algorithm with the `StringToWordVector` filter and the `WordTokenizer` filter—which divides the text into words—to detect code-switching between the Egyptian dialect and English. We achieved an accuracy of 55%, an average recall of 55%, an average precision of 67% and an average F-measure of 44%. Table 4.1 shows the results of this experiment.

After that, we examined the SVM classifier with the `CharacterNGramTokenize` filter, which divides text into n-grams. We tried four different types of n-grams: `UniGram`, `BiGram`, `TriGram`and `FourGram`.Table 4.1 shows that `UniGram` achieved an accuracy of 97.5%, which is a higher accuracy rate than the other n-grams models.

4.2.2 Detecting Code-switching between the Egyptian dialect, Saudi dialect and English

The second experiment was conducted to evaluate the automatic detection of code-switching among the Egyptian dialect, Saudi dialect and English. As in the first experiment, the testing text was extracted from the BAEC corpus for cases containing only the Egyptian dialect, Saudi dialect and English, that is, around 18,957 words and 85,099 characters.

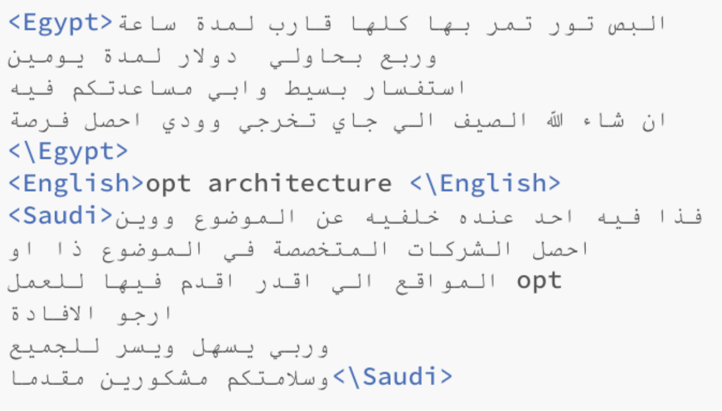
Using PPM produced an accuracy of 97.8%, an average recall of 89.9%, an average precision of

97.7% and an average F-measure of 93.2%. Table 4.2 shows the results of this experiment.

Figure 4.3 illustrates some Saudi dialect was predicted as Egyptian dialect, such as الصيف الي جاي تخرجي وودي احصل فرصة , and some Saudi dialect was correctly predicted as Saudi dialect, such as اوالمواقع الي اقدر اقدم فيها للعمل. Also, the word '*opt*', an abbreviation in architecture, was predicted as a Saudi dialect word. In this testing text, several English abbreviations were predicted as Saudi dialect or Egyptian dialect words, such as *HR*, *IBDL*, *PMP*, *CTRL C* and so on. The results of the first and second experiments show a clear need to build a new English training corpus that contains abbreviations.

Table 4.2: The results of the second experiment.

	Accuracy	Precision	Recall	F-Measure
PPM	**0.978**	**0.977**	**0.899**	**0.931**
SVM (WordTokenizer)	0.482	0.571	0.482	0.378
SVM (UniGram)	0.807	0.862	0.807	0.825
SVM (BiGram)	0.718	0.771	0.718	0.724
SVM (TriGram)	0.567	0.628	0.569	0.535
SVM (FourGram)	0.481	0.546	0.481	0.406



```
<Egypt>البص تور تمر بها كلها قارب لمدة ساعة
وربع بحاولي  دولار لمدة يومين
استفسار بسيط وابي مساعدتكم فيه
ان شاء الله الصيف الي جاي تخرجي وودي احصل فرصة
<\Egypt>
<English>opt architecture <\English>
<Saudi>فذا فيه احد عنده خلفيه عن الموضوع ووين
احصل الشركات المتخصصة في الموضوع ذا او
opt المواقع الي اقدر اقدم فيها للعمل
ارجو الافادة
وربي يسهل ويسر للجميع
وسلامتكم مشكورين مقدما<\Saudi>
```

Figure 4.3: An example of the confusion between the Saudi and Egyptian dialects from the second experiment's output using PPM.

Repeating the second experiment by using the SVM classifier with the `StringToWordVector` filter and the `WordTokenizer` filter, produced an accuracy of 48.2%, an average recall of 48.2%, an average precision of 57.1% and an average F-measure of 37.8%. Table 4.2 shows the results of this experiment. We also examined the SVM classifier with the `CharacterNGramTokenize` filter and tried different types of n-grams, as shown in Table 4.2. Table 4.2 shows that `UniGram` achieved an accuracy of 80.7%, which is a higher accuracy rate than the other n-grams models.

4.2.3 Detecting Code-switching between the Egyptian dialect, Saudi dialect, MSA and English

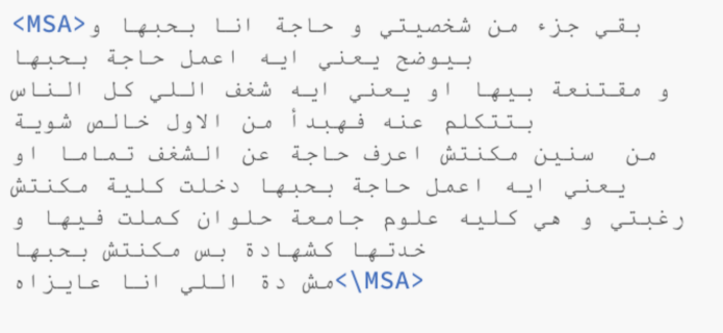
The third experiment was considered a more complex task for the PPM classifier because it had four different classes: the Egyptian dialect, the Saudi dialect, MSA and English. The testing file, which was around 5,002 words and 23,668 characters, was also manually extracted from the BAEC corpus containing the Egyptian dialect, the Saudi dialect, MSA and English.

Using PPM for the third experiment obtained an accuracy of 53.261%, an average recall of 53.9%, an average precision of 56.2% and an average F-measure of 55.1%. Table 4.3 shows the results of this experiment.

Table 4.3: The results of the third experiment.

	Accuracy	Precision	Recall	F-Measure
PPM	0.532	0.562	0.539	0.551
SVM (WordTokenizer)	0.263	0.351	0.263	0.239
SVM (UniGram)	**0.602**	**0.680**	**0.602**	**0.597**
SVM (BiGram)	0.511	0.611	0.511	0.421
SVM (TriGram)	0.371	0.553	0.371	0.352
SVM (FourGram)	0.295	0.446	0.294	0.263

Figure 4.4 shows a sample of the confusion between the Egyptian dialect and MSA in the third experiment. All these sentences are in the Egyptian dialect but were predicted as MSA. We speculate that the reason for this disappointing result was that the MSA corpus used to train the MSA model did not represent the MSA found on Facebook because it was built from news websites. To prove this, we examined the overall compression code lengths of the sample marked-up text for the different model configurations, as shown in Table 4.3.



```
بقي جزء من شخصيتي و حاجة انا بحبها و<MSA>
بيوضح يعني ايه اعمل حاجة بحبها
و مقتنعة بيها او يعني ايه شغف اللي كل الناس
بتتكلم عنه فهبدأ من الاول خالص شوية
من  سنين مكنتش اعرف حاجة عن الشغف تماما او
يعني ايه اعمل حاجة بحبها دخلت كلية مكنتش
رغبتي و هي كليه علوم جامعة حلوان كملت فيها و
خدتها كشهادة بس مكنتش بحبها
<\MSA>مش دة اللي انا عايزاه
```

Figure 4.4: Sample of some Egyptian dialect sentences predicted as MSA from the third experiment's output using PPM.

Table 4.4: Minimum code lengths for different models.

Different models used to segment the text	Min. code length (bits)
Egyptian, Saudi, MSA and English models	262977.688
Egyptian, Saudi and English models	258722.891
Egyptian and English models	261998.099

Table 4.4 shows that the Egyptian, Saudi and English models have the lowest minimum code lengths, with 258722.891 bits, so these would be the more appropriate models for this testing file. Adding a fourth model—MSA—to these models resulted in an increase of the minimum code length.

The most suitable solution to overcome this issue was to build a new MSA Facebook corpus trained on MSA text specifically taken from Facebook because this would be very different compared with the MSA corpus (which is mainly about politics) used in this experiment. Table 25 illustrates the differences between the MSA used in the experiment and MSA used in Facebook. The MSA used in the experiment was mainly about politics as opposed to social life. However, because of time limitations, we were unable to build a new MSA Facebook corpus.

Table 4.5: The differences between the MSA used in the experiment and MSA used on Facebook.

	Example
MSA as used in the experiment	في مجال حقوق الإنسان والدعوة إلى إنشاء هياكل هياكل أساسية وطنية في مجال حقوق الإنسان والاضطلاع بأنشطة وعمليات ميدانية فيما يتصل بحقوق الإنسان
MSA as used on Facebook	وقد وضع في الاعتبار ان الجميع سيكون مشغول في اليوم الثاني من العيد حتى يتمكن الاطفال من الذهاب لمدارسهم ويتمكن أولياء امورهم من الذهاب للعمل بالنسبه للرجال سيكون اجتماعهم المسائي في مقر النادي السعودي من الثامنه وحتى مابعد وليمة العشاء نلقاكم على خير وكل عام وأنتم بخير

Repeating the third experiment using the SVM classifier with the `StringToWordVector` filter and `WordTokenizer` filter obtained an accuracy of 26.3%, an average recall of 26.3%, an average precision of 35.1% and an average F-measure of 23.9%. Table 19 shows the results of this experiment. We then examined the SVM classifier with the `CharacterNGramTokenize` filter and tried different types of n-grams, as shown in Table 4.5. Table 4.5, which shows that `UniGram` achieved an accuracy of 60.2%, which is a higher accuracy rate than the other n-grams models.

4.3 Conclusion to this Chapter

This chapter aimed to answer the following research question: how does the effectiveness of the PPM compression-based approach compare against traditional CML algorithms for detecting code-switching in varieties and dialects from social media platforms? So, we have compared the CML classifier SVM and PPM compression-based approach with the automatic detection of code-switching in Arabic text. Our experiments have shown that PPM can achieve a higher accuracy rate than SVM when the training corpus correctly represents the language or dialect under study. When this condition is satisfied, the compression-based approach will be more effective for automatically detecting code-switching in written Arabic text. Second, when Arabic and English are classified using SVM, the `CharacterNGramTokenize` filter is a more appropriate filter to use than the `WordTokenizer` filter because the difference between these two languages is best modelled using characters. Third, the `CharacterNGramTokenize` filter is also more appropriate for a comparison between SVM and PPM because PPM is a character-based model.

The first experiment focused on the detection of code-switching between the Egyptian dialect and English. PPM obtained an accuracy of 99.8% on testing data from the BAEC corpus, which was 2.3% higher than the SVM classifier's accuracy. The second experiment investigated the automatic detection of code-switching among the Egyptian dialect, the Saudi dialect and English. PPM achieved an accuracy of 97.8%, which was 17.1% higher than the SVM classifier. Finally, the third experiment detected code-switching among the Egyptian dialect, the Saudi dialect, MSA and English. The SVM classifier obtained an accuracy of 60.2%, which was 6.9% higher than PPM.

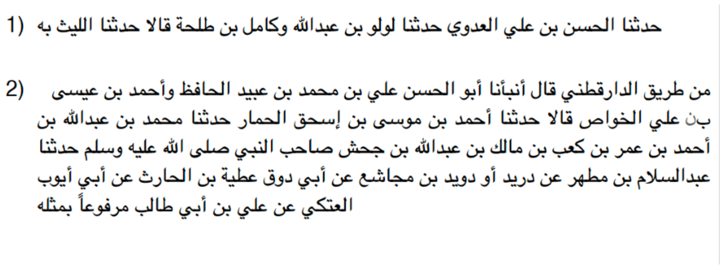
Clearly, the MSA corpus used to train the MSA PPM model in the third experiment did not represent MSA text in Facebook because it was built from news websites. As part of future work, a possible solution to overcome this issue would be to build a new MSA Facebook corpus trained on MSA text specially taken from Facebook. In addition, distinguishing between MSA and Arabic dialects is very difficult because most Arabic users, especially Saudis and Egyptians, mix MSA with their dialects. Finally, the use of a new English corpus containing all the possible abbreviations should further improve the results.

In the following chapter, we examine the effect of the PPM, CML and DL methods in classifying Arabic Hadith into its various components.

Chapter 5

Hadith Components Categorisation

Most Hadith segmentation studies have segmented Hadiths into Isnads and Matans without studying whether the Hadiths needed to be segmented or not. The reason for is that researchers used the six canonical Hadith books (well-structured Hadith books). Most Hadiths in these books have their two main components. Also, In the early Islamic period, Muslim scholars did not cite Hadiths without mentioning their Isnads. Today, however, most cite Hadiths in their speech, articles or books without providing their Isnads (J. A. Brown 2009). Hence, some Hadith books mention the Matan without Isnad, such as الأحاديث المجموعة في الأحاديث الموضوعة and الجد الحثيث في بيان ماليس بحديث, and some Hadith books mention some Isnads without Matans (see Figure 5.1). Thus, Hadiths with one component did not need to be segmented



1) حدثنا الحسن بن علي العدوي حدثنا لولو بن عبدالله وكامل بن طلحة قالا حدثنا الليث به

2) من طريق الدارقطني قال أنبأنا أبو الحسن علي بن محمد بن عبيد الحافظ وأحمد بن عيسى
بن علي الخواص قالا حدثنا أحمد بن موسى بن إسحق الحمار حدثنا محمد بن عبدالله بن
أحمد بن عمر بن كعب بن مالك بن عبدالله بن جحش صاحب النبي صلى الله عليه وسلم حدثنا
عبدالسلام بن مطهر عن دريد أو دويد بن مجاشع عن أبي دوق عطية بن الحارث عن أبي أيوب
العتكي عن علي بن أبي طالب مرفوعاً بمثله

Figure 5.1: Some examples of Isnad without Matan.

To the best of our knowledge, no study has concentrated on classifying a specific Hadith text based on Isnad, Matan and full Hadith (Isnad and Matan) as a first step before segmenting Hadith into its two main parts. Figure 5.2 shows an overview of the pipeline for segmenting Hadiths. The aim of this chapter is to fill this research gap.

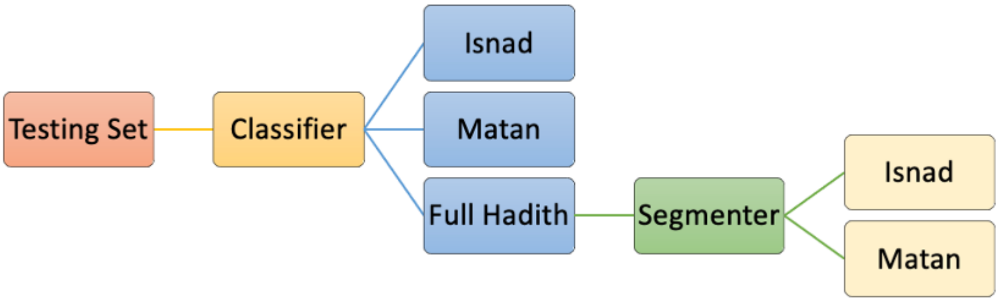


Figure 5.2: Overview of the pipeline for segmenting Hadiths.

In this chapter, we aim to achieve a few goals. First, we aim to build DL classifiers. Then, we examine three different methods—DL, compression based and CML—on two different Hadith corpora to discover the most effective method of classifying Arabic Hadiths into Isnad, Matan and full Hadith. In addition, we compare the execution time (training and testing time) of these three methods to discover the faster method of classifying Arabic Hadiths.

Part of this chapter is based on the following paper:

Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.

The chapter is structured as follows: section 5.1 discusses the proposed DL models that were applied for the automatic classification of Isnad, Matan and full Hadith; section 5.2 describes the experimental setup for Hadith components classification. Next, section 5.3 describes the character-based categorisation and word-based categorisation experiments and reports the experimental results. Finally, section 5.4 concludes the chapter.

5.1 Proposed Deep Learning Models

We have used the CNN and LSTM basic models, here proposing a hybrid model of them. The CNN–LSTM hybrid model incorporates the advantages of both CNN and LSTM. CNN can capture the local features of the text. However, for long sequences of words, CNN cannot preserve long-term dependencies. LSTM overcomes this vanishing gradient problem by capturing long-term dependencies in a long sequence of words (J. Zhang et al. 2018; X. Li and Ning 2020). J. Zhang et al. (2018) reported that this hybrid model can enhance the accuracy rate of text classification.

The basic hierarchy of our DL models consist of an input embedding layer, a hidden layer and a dense output layer. Our goal is to build simple DL models to compare them against CML models. Word embeddings are standard representation of word meanings used in NLP (Jurafsky and Martin 2008). The embedding layer is an important layer for DL models because it captures the relationships between words that would be hard to capture otherwise. In this layer, each word in the input data is represented by a dense vector of fixed size. We used this layer to learn an embedding for all of the words in our training datasets.

The dense output layer takes the number of classes as its output dimension. Because it is a multi-classification problem, the `Softmax` function was used as the activation function.

Our DL model's hierarchy is as follows:

- **CNN:** The architecture of our CNN model consists of one CNN layer with 15 filters and a kernel size of 3, which is followed by global max pooling with default values (see Figure 5.3).
- **LSTM:** The architecture of our LSTM model consists of one LSTM layer with `hidden_nodes=15` and `return_sequences=True`.The `return_sequences` argument returns all the outputs of the hidden states of each time steps. The next layer is global max pooling with default values (see Figure 5.3).
- **CNN-LSTM:** The architecture of this model consists of one CNN and one LSTM, which is followed by global max pooling with default values (see Figure 5.4)

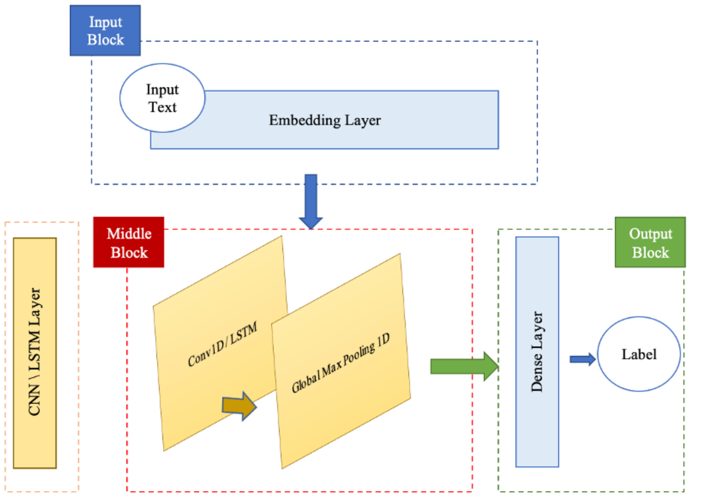


Figure 5.3: The architecture of our CNN and LSTM models.

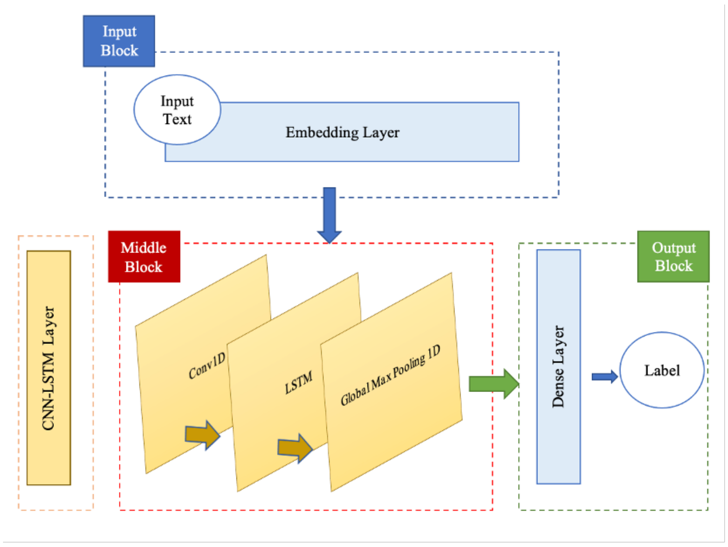


Figure 5.4: The architecture of CNN-LSTM model.

When fitting the DL models, we used the `callbacks` function with `early_stop` method to monitor the performance of our model. This method can end the training process if the accuracy stops improving. In addition, we added the `patience` argument with four epochs to delay the early stopping for a number of epochs with no improvement.

The DL hyper-parameters have been setup by using the manual search method, which is based on researcher observation and judgment. During the initial experiments, we trained our model before evaluating the accuracy rate and tweaking the hyper-parameters based on that result. Then, we repeated the process of changing hyper-parameters again until we achieved a satisfactory accuracy.

We ran our DL models on Google Colab. The main advantage of Google Colab is that it provides three types of runtime: central processing unit (CPU), graphical processing unit (GPU) and tensor processing Unit (TPU). First, we used CPU to train our model, but this was quite slow (one experiment took more than three hours). Hence, in our experiments, we used GPU to accelerate the execution time, which brought the time down to a few minutes.

5.2 Experimental Setup

For the Hadith components categorisation, we selected two different Hadith corpora: a NAH corpus (Chapter 3) and the Leeds University and King Saud University Hadith corpus (LK). The main advantages of these two corpora are that they are freely available to the Hadith research community and have different Hadith structures.

Our experiments used two datasets (*N3_1* and *N3_2*) from the NAH corpus. Additionally, they used the *Sahih Al-Bukhari* and *Sahih Muslim* datasets from the LK corpus. We automatically extracted Isnads, Matans and full Hadiths from NAH and LK for training models and testing purposes. In addition, our experiments used word-based and character-based features from one to 8-grams.

The goal was to classify each Hadith record into its right class. Each Hadith record was assigned one of the following three classes:

- <**Isnad**> records that contain just Isnads without Matans.
- <**Matan**> records that contain just Matans without Isnads.
- <**FullHadith**> records that contain both Isnads and Matans.

To guarantee a balanced distribution in all classes, we limited the number of records per class. Thus, the training phases used an equal number of records per category. For example, from *N3_1*, we used 1,264 Isnad records, 1,264 Matan records and 1,264 full Hadith records. Table 5.1 summarises the number of records per class in each dataset used in this study.

Table 5.1: Summary of the datasets used.

Datasets	No. of records per class
N3_1	1,264
N3_2	1,498
Sahih Al-Bukhari	7,340
Sahih Muslim	6,789

After labelling each record according to the Hadith components and limiting the number of records per class, all records were concatenated into three groups (Isnad/Matan/full Hadith) in the training process. Therefore, three models were produced. Then, in the testing process, each record was classified into Isnad, Matan or full Hadith. The following diagram demonstrates the classification procedure:

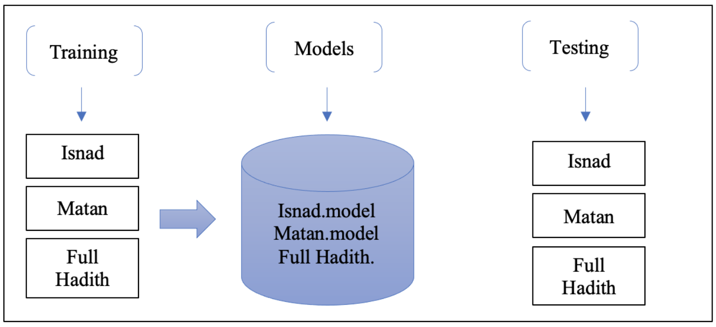


Figure 5.5: Overview of the classification procedure used.

5.2.1 Character-based Categorisation Approach

We applied character-based categorisation approaches: PPM and CML algorithms. After finding the best order in section 6.3.1, we applied PPM classification using order 7. Also, we applied different CML classifiers with different character n-grams.

5.2.2 Word-based Categorisation Approach

The CML algorithms were applied using word n-grams. Also, we applied DL algorithms to find the best result. Then, we compared the word-based approaches with the results generated from the PPM character-based approach.

5.3 Experiments and Results

Different experiments were performed to evaluate the compression-based classifier (Teahan 2018), CML classifiers such as the SVM, NB and DT classifiers, including the LSTM network and CNN classifiers, for the automatic classification of Isnads, Matans and full Hadiths in Arabic Hadith texts.

5.3.1 Comparing CML Classifiers with Different Features

Our aim was investigated using different CML classifiers implemented in Weka (Hall et al. 2009) with default parameters. We applied forth main experiments and each experiment had two subexperiments (word and character). First, we built the vector list by using the `StringToWordVector` filter. Also, we applied the `WordTokenizer` filter and `CharacterNGramTokenize` filter for word-based and character-based features, respectively. The `WordTokenizer` filter divides the text into words, and the `CharacterNGramTokenize` filter divides text into n-grams. We noticed that adding more character n-grams resulted a low accuracy rate so, we did not add more (see Table 5.2).

In the first experiment, *N3_1* was selected as a training set, and *Sahih Al-Bukhari* as a testing set. The NB classifier with the word unigram feature achieved the best results with an accuracy of 92.67%. The DT classifier with the character 8-grams feature achieved the worst result, with an accuracy of 69.9%. In the second experiment, was selected *Sahih Al-Bukhari* as a training set, and *N3_1* as a testing set. The SVM classifier with a character 5-grams feature performed the best, here with an accuracy of 83%, and the DT classifier with a character 8-grams feature performed the worst, here with an accuracy of 48%.

Also, in the third experiment, *N3_2* was selected as a training set, and *Sahih Muslim* as a testing set. We found that the NB classifier with a character 5-grams feature achieved the highest score (88.9%) when compared with the other classifiers. At the same time, it had the lowest score (66%) when we changed the feature to be a character unigram feature. In the fourth experiment, *Sahih Muslim* was selected as a training set, and *N3_2* as a testing set. The NB classifier with the word unigram feature was found to perform well, with results as high as 78%. Also, it performed poorly with a character 8-grams feature, showing an accuracy of 46.9%.

In general, Table 5.2 shows that, first, the CML classifiers reported the best results when the NAH corpus was used as a training set and the LK corpus as a testing set and vice versa. Hence, the accuracy range for the first and second experiments were from 92% to 69.9% and from 83% to 48%, respectively. Subsequently, using the NAH corpus to train the models gave the best results. By comparing the third and fourth experiments, we found that the accuracy range for the third experiment was between 88.9% and 66%, while the accuracy range for the fourth experiment was between 78% and 46.9%, which showed that using the NAH corpus to train the models gave the best results. Second, the SVM and NB classifiers obtained higher results than the DT classifier.

Table 5.2: Performance evaluation of the CML classifiers using word and character features. Each experiment revealed the accuracy of the CML classifiers trained and tested on different datasets. The highest accuracy rates are shown in bold.

	Dataset		Training set: N3_1 Testing set: Sahih Al-Bukhari		
	Classifiers		SVM	NB	DT
First Experiment	Word	Unigrams	90.36	**92.67**	89.97
	Character	Unigrams	**79.69**	73.98	76.11
		Bigrams	**90.27**	86.19	89.28
		Trigrams	89.99	**90.49**	87.49
		4-grams	88.91	**91.87**	88.71
		5-grams	87.68	**91.79**	89.57
		6-grams	84.71	**89.54**	80.83
		7-grams	80.88	84.55	**85.32**
		8-grams	**77.85**	73.75	69.93
	Dataset		Training set: Sahih Al-Bukhari Testing set: N3_1		
	Classifiers		SVM	NB	DT
Second Experiment	Word	Unigrams	76.43	**74.45**	66.67
	Character	Unigrams	**69.06**	65.42	61.49
		Bigrams	**83.29**	71.81	71.02
		Trigrams	**82.73**	76.37	72.12
		4-grams	**81.07**	75.66	68.11
		5-grams	**81.44**	68.91	71.36
		6-grams	**78.51**	58.77	75.34
		7-grams	**78.93**	51.64	63.63
		8-grams	**74.87**	49.72	48.35
	Dataset		Training set: N3_2 Testing set: Sahih Muslim		
	Classifiers		SVM	NB	DT
Third Experiment	Word	Unigrams	82.16	**86.83**	82.28
	Character	Unigrams	**77.68**	66.15	69.14
		Bigrams	83.96	**84.61**	79.80
		Trigrams	86.50	**86.69**	79.71
		4-grams	85.45	**87.99**	79.75
		5-grams	83.56	**88.94**	76.66
		6-grams	81.14	**88.27**	72.13
		7-grams	82.55	**84.87**	78.28
		8-grams	77.66	**78.31**	77.65
	Dataset		Training set: Sahih Muslim Testing set: N3_2		
	Classifiers		SVM	NB	DT
Forth Experiment	Word	Unigrams	78.35	**78.53**	74.86
	Character	Unigrams	**68.17**	67.34	66.30
		Bigrams	**78.44**	73.38	72.67
		Trigrams	**77.90**	77.17	67.68
		4-grams	75.30	**77.81**	65.41
		5-grams	74.40	**74.94**	62.42
		6-grams	72.78	**72.91**	69.97
		7-grams	71.26	71.44	**72.40**
		8-grams	**69.77**	46.92	62.98

5.3.2 Comparing the PPM, CML and DL classifiers

The next experiments were performed to (1) compare the DL, CML and PPM classifiers, (2) compare the execution time (training and testing time) of these three methods to discover the faster method of classifying Arabic Hadiths. A comparison of the processing times and classification accuracy for all the algorithms under study are summarised in Table 5.3.

Table 5.3: A comparison of the processing times and classification accuracy for all the algorithms under study.

				Testing Set							
	Training Set	Classifier		N3_1		N3_2		Sahih Al-Bukhari		Sahih Muslim	
				Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
		PPM	PPM			0.8493	22	0.7025	40	0.7393	35
			SVM			0.9147	**3**	0.9036	**4**	0.8412	**4**
		CML	NB			0.8862	4	0.9267	14	0.8885	15
First Experiment	N3_1		DT (J48)			0.8873	17	0.8997	16	0.8312	16
			LSTM			0.9503	204	0.9533	303	**0.9077**	174
		DL	CNN			**0.9646**	26	0.9462	36	0.9038	29
			CNN-LSTM			0.9526	314	**0.9546**	155	0.9003	289
		PPM	PPM	0.8867	23			0.8813	36	0.7903	31
			SVM	0.9274	6			0.8936	**6**	0.8216	**6**
		CML	NB	0.8818	**4**			0.9157	16	0.8683	16
Second Experiment	N3_2		DT (J48)	0.8889	22			0.8726	22	0.8228	24
			LSTM	**0.9596**	50			**0.9705**	18	0.9074	45
		DL	CNN	0.9464	13			0.9625	61	0.8811	14
			CNN-LSTM	0.9451	21			0.9487	92	**0.9207**	44
		PPM	PPM	0.8476	21	0.8554	22			0.8577	29
			SVM	0.7643	41	0.7076	41			0.8593	44
		CML	NB	0.7445	**11**	0.6915	**10**			0.9031	**20**
Third Experiment	Sahih Al-Bukhari		DT (J48)	0.6667	169	0.6434	170			0.8804	166
			LSTM	0.9192	192	0.8938	122			0.9074	45
		DL	CNN	0.9171	76	**0.9296**	65			0.9381	29
			CNN-LSTM	**0.9264**	78	0.9201	259			**0.9425**	150
		PPM	PPM	0.8236	17	0.8095	17	0.9165	28		
			SVM	0.8044	142	0.7835	139	0.9415	143		
		CML	NB	0.8224	**10**	0.7853	**10**	0.9557	21		
Fourth Experiment	Sahih Muslim		DT (J48)	0.7749	166	0.7486	159	0.9573	164		
			LSTM	**0.8852**	261	0.8357	201	**0.9807**	578		
		DL	CNN	0.7925	15	0.7878	15	0.9691	**18**		
			CNN-LSTM	0.8456	67	**0.8811**	265	0.9790	136		

In the first experiment, we trained the PPM models, CML models and DL models on the *N3_1* dataset and then tested each model on the *N3_2*, *Sahih Al-Bukhari* and *Sahih Muslim* datasets. When tested on *N3_2*, the CNN classifier achieved the highest score compared with the other classifiers at 96%. The PPM performed worse, with an accuracy of 84%. The CNN-LSTM obtained higher accuracy rates compared with other classifiers when testing the models on *Sahih Al-Bukhari* (95%). Also, the PPM performed worse, here with an accuracy of 70%. When tested on *Sahih Muslim*, , the LSTM classifier produced classification accuracy better than the other classifiers, specifically 90.77%. As before, the PPM obtained lower accuracy rates than the other classifiers in this experiment. As opposed to what was expected, the LSTM and CNN-LSTM classifiers consumed more time and the SVM less when it came to the execution time.

In the second experiment, we trained the five models on the *N3_2* dataset and then tested them on the *N3_1*, *Sahih Al-Bukhari* and *Sahih Muslim* datasets. When tested on *N3_1*, the LSTM classifier achieved the highest score compared with the other classifiers, with a 95.9% accuracy. The NB classifier performed worse, with an accuracy of 88.18%. The LSTM obtained higher accuracy rates compared with other classifiers when testing the models on the *Sahih Al-Bukhari* (97%). The DT classifier performed worse, with an accuracy of 87%. When tested on *Sahih Muslim*, the CNN-LSTM classifier reported a higher accuracy rate than the other classifiers (92%). As before, the PPM obtained lower accuracy rates (79%) than the other classifiers in this experiment. The SVM and NB classifiers were able to generate an output with a lower classification accuracy and a much shorter execution time compared with the LSTM and CNN-LSTM classifiers and vice versa.

In the third experiment, we trained the five models on the *Sahih Al-Bukhari* dataset and then tested the models on the *N3_1*, *N3_2* and *Sahih Muslim* datasets. When tested on *N3_1*, the CNN-LSTM classifier achieved the highest score compared with the other classifiers (92%). The DT classifier performed worse, with an accuracy of 66%. The CNN obtained higher accuracy rates compared with the other classifiers when testing the models on the *N3_2* (92.9%). Also, the DT classifier performed worse, with an accuracy of 64%. When tested on *Sahih Muslim*, the CNN-LSTM classifier reported a higher accuracy rate compared with the other classifiers (94%). The PPM obtained lower accuracy rate (85.77%) than the other classifiers in this experiment. The SVM consumed less time and the LSTM and CNN-LSTM classifiers consumed more when it came to the execution time.

In the last experiment, we trained the four models on the *Sahih Muslim* dataset and then tested the models on the *N3_1*, *N3_2* and *Sahih Al-Bukhari* datasets. When tested on *N3_1*, the LSTM classifier achieved the highest score compared with the other classifiers (88%). The DT classifier performed worse, with an accuracy of 77%. The CNN-LSTM obtained a higher accuracy rate (88%) compared with other classifiers when testing the models on the *N3_2*. Also, the DT classifier performed worse, with an accuracy of 74%. When tested on *Sahih Al-Bukhari*, the LSTM classifier reported a higher accuracy rate than the other classifiers (98%). The PPM obtained lower accuracy rates (91%) than the other classifiers. As the previous experiment, the SVM took less time and the LSTM and CNN-LSTM classifiers consumed more when it came to the execution time.

For the first and second experiments, we noticed that the CML classifiers reported a higher accuracy rate than the PPM classifier when training on the *N3_1* and *N3_2* datasets and testing them on different Hadith structures (*Sahih Al-Bukhari* and *Sahih Muslim*). In contrast, when the training size was big such as in the third and the final experiments, we noticed that the PPM classifier obtained a higher accuracy rate than the CML classifiers when training on the *Sahih Al-Bukhari* and *Sahih Muslim* datasets and testing them on different Hadith structures (*N3_1* and *N3_2*).

From our experiments, we first observed that the classification results for the PPM classifier were enhanced when training the models on *Sahih Al-Bukhari* and *Sahih Muslim*, this may it be because of the large training size. Figure 5.6 demonstrates how the performance of DL and CML classifiers were not affected by the training size; they performed well when training the models on *N3_1* and *N3_2*, which are a smaller datasets than *Sahih Al-Bukhari* and *Sahih Muslim*, which contrasts the performs of PPM. It also shows that the DL and CML classifiers reported the best results when the NAH corpus was used as a training set and the LK corpus as a testing set and vice versa, which means that the NAH corpus well represents the Hadith's components. Third, the DL classifiers produced classification accuracy better than the other classifiers, but the execution time was high, even though they were using GPU hardware. Finally, by comparing DL classifiers, the CNN took less time to execute. By comparing the CML classifiers, the SVM and NB classifiers took less time to execute than the DT classifier. The PPM classifier was faster than DL classifiers. By comparing all the algorithms, CML classifiers were found to be faster.

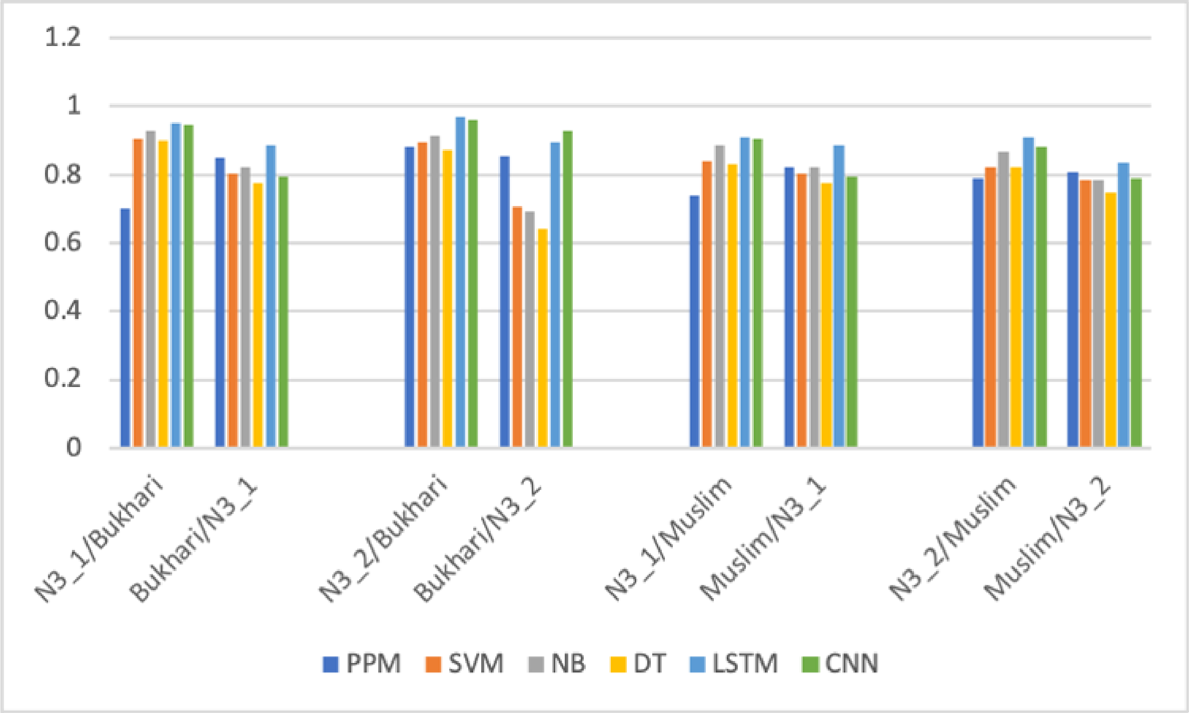


Figure 5.6: A comparison of the accuracy rate when using the NAH corpus as a training set and the LK corpus as a testing set and vice versa (training set/testing set).

5.4 Conclusion

This chapter aimed to answer the following research question: how does the effectiveness of the PPM compression-based approach compare against classical ML and DL algorithms for classifying Arabic Hadith text according to Hadith components? So, we compared the DL classifiers—LSTM, CNN and CNN-LSTM—the PPM compression-based classifier and the CML classifiers—SVM, NB and DT—for the automatic classification of Arabic Hadith text into Isnad, Matan and full Hadith. This was the first step towards segmenting a Hadith into its two main components.

First, our experiment showed that DL classifiers achieved a higher classification accuracy compared with the other classifiers under study. However, the DL classifiers took more time to execute, even though they were using GPU hardware. Second, we observed that CML classifiers behaved differently when trained and tested on different structures of Arabic Hadith text. Third, there was a possibility that the results of the PPM classifier were affected by the training size. Finally, earlier, we mentioned that J. Zhang et al. (2018) reported that the CNN-LSTM hybrid model enhanced the accuracy rate of text classification. However, our experiments showed that, in most cases, the results of the LSTM classifier were better than the

CNN-LSTM classifier's results.

One major drawback of DL classifiers using Tensorflow and keras is that everything is hidden so that classification error logs are difficult to analyse.

In the following chapter, we examine the effect of the PPM method in segmenting Arabic Hadiths into its components.

Chapter 6

Hadith Segmentation and Authenticity Classification

Automated text segmentation is the task of building a tool that can automatically identify sentence boundaries in a given text and divide them into their components. Converting unstructured text into a structured format is especially important when dealing with unstructured text, such as web text or old documents. One of the most important types of old holy Islamic texts in the Arabic language is the Hadith.

Automatic Hadith segmentation of Isnads and Matans can help Hadith researchers, some of whom focus on an Isnad with the aim of studying the narrator's reliability, the links between them or how a specific Hadith has been transferred over time, sometimes generating a graphical visualisation to represent this (Azmi and Bin Badia 2010). Other research concentrates on Matans to classify Hadiths into topics (Saloot et al. 2016).

Teahan (2000) used PPM to solve several NLP problems, such as text classification and segmentation. Altamimi and Teahan (2017) pointed out that using a character-based compression scheme for tasks, such as detecting code-switching and gender/authorship categorisation, is more effective than word-based CML approaches. Many current Hadith studies have used a word-based method to segment Hadith from the six canonical Hadith books, but the method that this chapter uses is a character-based PPM compression method, which can automatically segment Isnads and Matans. Our goals are to evaluate PPM segmenter on (1) unstructured Hadith text from lesser-known Hadith books and (2) well-structured Hadith text from the six canonical Hadith books.

On the other hand, the process of distinguishing authentic Hadiths from non-authentic Hadiths is the task of Hadith judgement science. Najeeb (2021) reported that Hadiths can be automatically judged using a computerised classifier, such as CML and DL classifiers, which can assist Hadith researchers.

In computer science, Hadith authentication refers to the classification of Hadiths as authentic or non-authentic using artificial intelligence methods. Several types of NLP methods can be applied to solve the problem of Hadith authentication, such as CML, DL and PPM. However, very few studies have been published in this area. Most have focused on Isnads for automatically judging the authenticity of Hadiths; a paper by Hakak et al. (2020) indicated that authentication based on Matans is one challenge facing the authenticating of digital Hadiths because it requires to verify the Matan on the basis of Quran. Also, the research has examined the use of rule-based (Ghazizadeh et al. 2008) or CML classifiers (Najiyah et al. 2017) to automatically detect a Hadith's authenticity, yet none examined the use of DL or PPM classifiers. This helped to inspire the work described in this chapter.

Part of this chapter is based on the following papers:

Tarmom T; Atwell E; Alsalka MA (2022). Deep Learning vs Compression-Based vs Traditional Machine Learning Classifiers to Detect Hadith Authenticity. In Annual International Conference on Information Management and Big Data, pp. 206–222. Springer, Cham.

Tarmom T; Atwell E; Alsalka MA (2020). Automatic Hadith Segmentation using PPM Compression. In Proceedings of the 17th International Conference on Natural Language Processing (ICON), pp. 22–29.

The remainder of the chapter is structured as follows: section 6.1 describes the character-based segmentation method; section 6.2 explains the experimental setup and data used in the study; section 6.3 discusses the results of Hadith segmentation experiments; section 6.4; section 6.5 discusses the goals of Hadith authenticity experiments; section 6.6 describes the experimental setup for this part; section 6.7 discusses the Hadith authenticity experiments and reports the results; and section 6.9 concludes the chapter and suggests future work.

6.1 Character-Based Segmentation Using PPM

We used the PPM, which is a character-based approach, to perform the segmentation. It utilises the Viterbi algorithm (Viterbi 1967), which uses a trellis-based search (Ryan and Nudd 1993) to find the segmentation with the best compression out of all possible segmentation search paths extended at the same time, discarding the poorly performing alternatives (Teahan 2018).

Figure 6.1 shows an illustrative example of the search tree for the text segmentation problem in the Tawa toolkit (Teahan 2018). In this example, the tree has a branching of two because two labels have been used: Isnad and Matan. The label <I> (used for the Isnad model) and <M> (used for the Matan model) show the transformed sequences within each node. If a character switches from one model to the other, the sentinel character is encoded. The compression code length is also calculated for the transformed sequence, which appears on the right of each node and below the last nodes. The smallest one, which is the best segmented one, is shown in bold font.

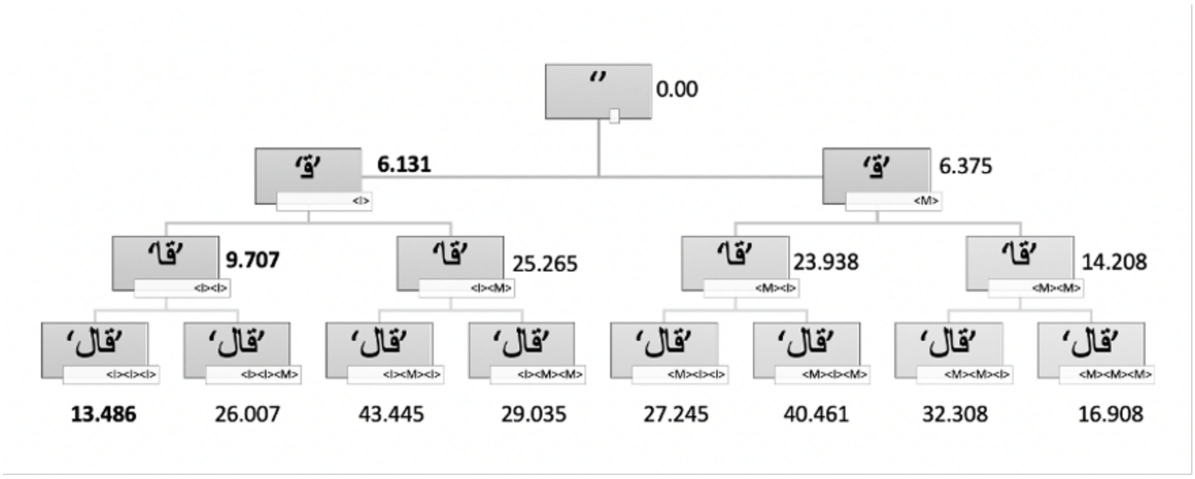


Figure 6.1: An illustrative example of the search tree for the text segmentation problem in the Tawa toolkit.

As we mentioned in Chapter 2 that character-based text compression methods have not been used in previous Hadith segmentation studies. Also, Teahan (2018) concluded that NLP tasks using the PPM character-based compression approach outperformed feature-based approaches, such as NB and SVM, so we chose the PPM approach for our experiments. Due to time limitation, we were unable to build another segmenter such as n-gram CML or DL segmenters and we did not find any freely available segmenter to compare it against the PPM segmenter.

6.2 Experimental Setup

Two experiments were performed as part of the evaluation of the compression-based method (provided by Tawa toolkit) (Teahan 2018) to automatically separate Hadiths into two components: Isnad and Matan. In the first experiment, we used two different datasets from the NAH corpus for training models and testing, and in the second experiment, we used the NAH corpus for training models and the Leeds University and King Saud University (LK) Hadith corpus Altammami et al. (2020), for testing PPM segmenter.

6.3 Experimental Results

This section describes in detail the experiments that were applied for this study.

6.3.1 First Experiment

In this experiment, the first dataset in the NAH corpus, *N1*, was chosen for training purposes. This dataset contains Hadith text from the *False, Disreputable, and Well-known Hadith Texts* book 'الأباطيل والمشاهير والمناكير للجورقاني'. It consists of 732 Hadiths and is 121,080 words long. Isnads and Matans were manually extracted from *N1* for the Isnad and Matan training models, which were 52,221 and 33,489 words long, respectively. The testing text was manually extracted from the third dataset in the NAH corpus, *N3_1*, which contains just Isnad and Matan and is 6,339 words long (see Table 6.1).

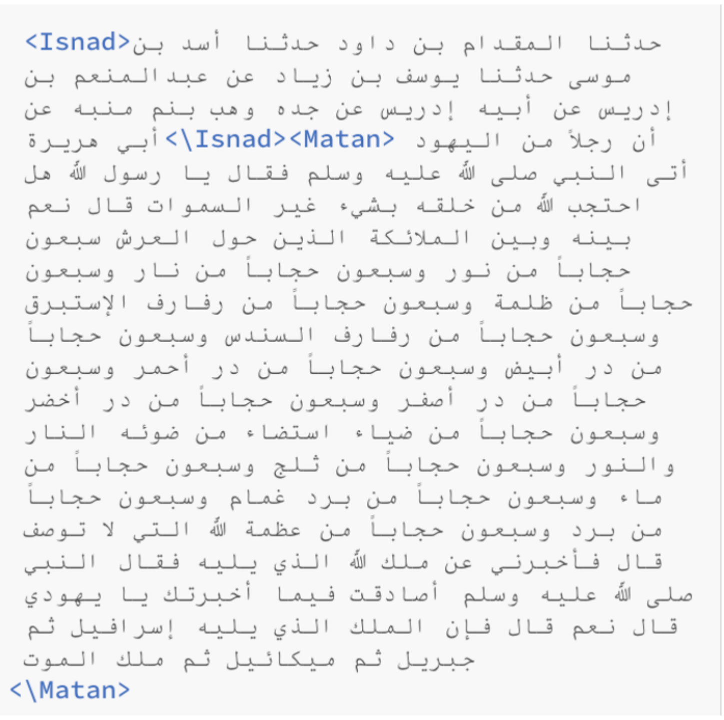
Table 6.1: Number of words for training and testing sets used.

Dataset	No. of words
Isnad training set	52,221
Matan training set	33,489
First experiment Testing set	6,339
Second experiment Testing set	10,539

For automatic Hadith segmentation, different orders of PPMD were performed, from order 2 to order 10. As shown in Table 6.2, order 7 obtained a higher accuracy of 92.76%, a higher average recall of 0.9365, a higher average precision of 0.9231, and a higher average F-measure of 0.9297. A sample output from the first experiment is shown in Figure 6.2. Figure 6.3 shows that the last part of Isnad texts were predicted as Matans, such as ‘عن جابر عن النبي صلى اللهّ عليه وسلم’ ‘*It has been narrated on the authority of Jabir on the authority of the Prophet, may God bless him and grant him peace*’ (highlighted in blue).

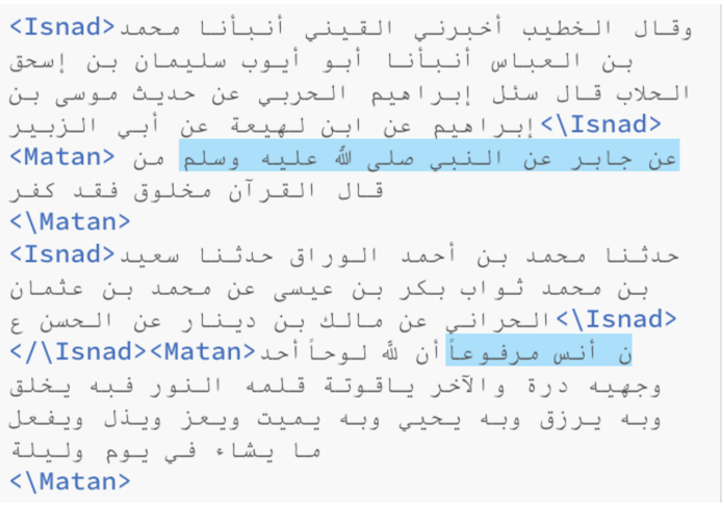
Table 6.2: Hadith segmentation using PPMD.

Orders	Accuracy (%)	Recall	Precision	F-measure
2	83.34	0.8580	0.8555	0.8568
3	87.20	0.8914	0.8801	0.8858
4	88.04	0.8996	0.8843	0.8919
5	87.02	0.8881	0.8800	0.8840
6	88.58	0.9022	0.8901	0.8961
7	**92.78**	**0.9365**	**0.9231**	**0.9297**
8	92.68	0.9356	0.9222	0.9288
9	92.67	0.9350	0.9215	0.9282
10	91.78	0.9275	0.9127	0.9200



```
<Isnad>حدثنا المقدام بن داود حدثنا أسد بن
موسى حدثنا يوسف بن زياد عن عبدالمنعم بن
إدريس عن أبيه إدريس عن جده وهب بنم منبه عن
أبي هريرة<\Isnad><Matan> أن رجلاً من اليهود
أتى النبي صلى ﷲ عليه وسلم فقال يا رسول ﷲ هل
احتجب ﷲ من خلقه بشيء غير السموات قال نعم
بينه وبين الملائكة الذين حول العرش سبعون
حجاباً من نور وسبعون حجاباً من نار وسبعون
حجاباً من ظلمة وسبعون حجاباً من رفارف الإستبرق
وسبعون حجاباً من رفارف السندس وسبعون حجاباً
من در أبيض وسبعون حجاباً من در أحمر وسبعون
حجاباً من در أصفر وسبعون حجاباً من در أخضر
وسبعون حجاباً من ضياء استضاء من ضوئه النار
والنور وسبعون حجاباً من ثلج وسبعون حجاباً من
ماء وسبعون حجاباً من برد غمام وسبعون حجاباً
من برد وسبعون حجاباً من عظمة ﷲ التي لا توصف
قال فأخبرني عن ملك ﷲ الذي يليه فقال النبي
صلى ﷲ عليه وسلم أصادقت فيما أخبرتك يا يهودي
قال نعم قال فإن الملك الذي يليه إسرافيل ثم
جبريل ثم ميكائيل ثم ملك الموت
<\Matan>
```

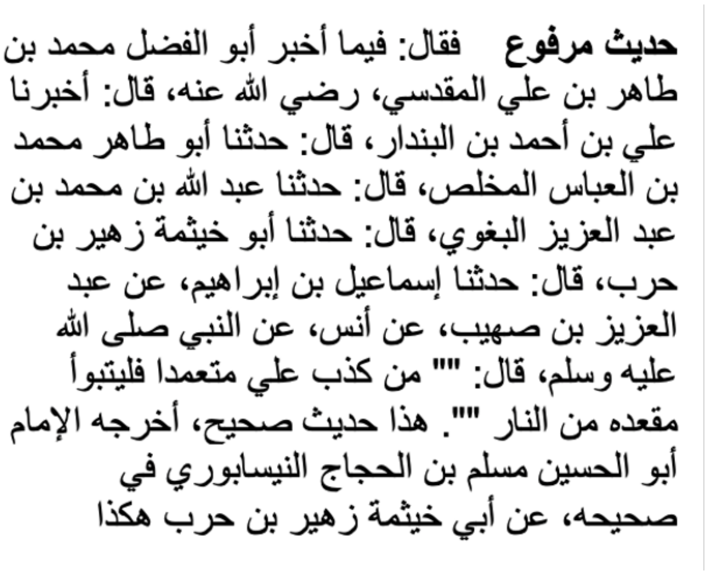
Figure 6.2: Sample output using PPMD with order 7.



```
وقال الخطيب أخبرني القيني أنبأنا محمد<Isnad>
بن العباس أنبأنا أبو أيوب سليمان بن إسحق
الحلاب قال سئل إبراهيم الحربي عن حديث موسى بن
<\Isnad>إبراهيم عن ابن لهيعة عن أبي الزبير
عن جابر عن النبي صلى الله عليه وسلم من <Matan>
قال القرآن مخلوق فقد كفر
<\Matan>
حدثنا محمد بن أحمد الوراق حدثنا سعيد<Isnad>
بن محمد ثواب بكر بن عيسى عن محمد بن عثمان
<\Isnad>الحراني عن مالك بن دينار عن الحسن ع
ن أنس مرفوعاً أن لله لوحاً أحد</\Isnad><Matan>
وجهيه درة والآخر ياقوتة قلمه النور فبه يخلق
وبه يرزق وبه يحيي وبه يميت ويعز ويذل ويفعل
ما يشاء في يوم وليلة
<\Matan>
```

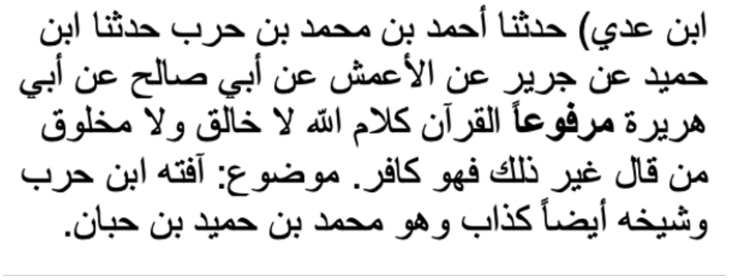
Figure 6.3: An example of confusion between an Isnad and Matan using PPMD with order 7.

We noticed that the structure of the Isnad texts used in the training set and testing set differed, creating some confusion in the results. The type of Hadith is given at the beginning of each Hadith in *N1*, for example, حديث مرفوع '*Marfo Hadith*', which is not labelled as Isnad (see Figure 6.4). In the *N*3_1 dataset, each type of Hadith has been written at the end of the Isnad (see Figure 6.5). Figure 6.6 shows that the Isnad and Matan were correctly predicted, but the word مرفوعا '*Marfo*' was wrongly predicted as belonging to a Matan because it did not appear in the Isnad training set (highlighted in blue).



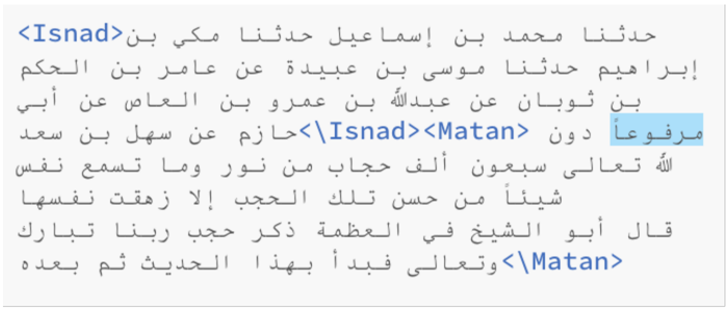
حديث مرفوع فقال: فيما أخبر أبو الفضل محمد بن
طاهر بن علي المقدسي، رضي الله عنه، قال: أخبرنا
علي بن أحمد بن البندار، قال: حدثنا أبو طاهر محمد
بن العباس المخلص، قال: حدثنا عبد الله بن محمد بن
عبد العزيز البغوي، قال: حدثنا أبو خيثمة زهير بن
حرب، قال: حدثنا إسماعيل بن إبراهيم، عن عبد
العزيز بن صهيب، عن أنس، عن النبي صلى الله
عليه وسلم، قال: "" من كذب علي متعمدا فليتبوأ
مقعده من النار "". هذا حديث صحيح، أخرجه الإمام
أبو الحسين مسلم بن الحجاج النيسابوري في
صحيحه، عن أبي خيثمة زهير بن حرب هكذا

Figure 6.4: An example of a Hadith from *N1* dataset (Hadith rank is in bold).



ابن عدي) حدثنا أحمد بن محمد بن حرب حدثنا ابن
حميد عن جرير عن الأعمش عن أبي صالح عن أبي
هريرة **مرفوعاً** القرآن كلام الله لا خالق ولا مخلوق
من قال غير ذلك فهو كافر. موضوع: آفته ابن حرب
وشيخه أيضاً كذاب وهو محمد بن حميد بن حبان.

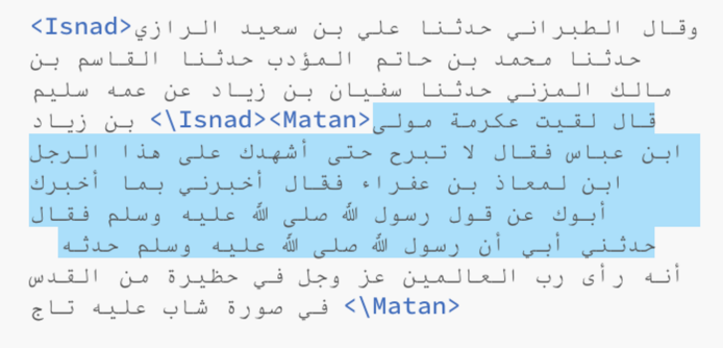
Figure 6.5: An example of a Hadith from $N3\_1$ dataset (Hadith rank is in bold).



<Isnad>حدثنا محمد بن إسماعيل حدثنا مكي بن
إبراهيم حدثنا موسى بن عبيدة عن عامر بن الحكم
بن ثوبان عن عبدالله بن عمرو بن العاص عن أبي
مرفوعاً دون <\Isnad><Matan>حازم عن سهل بن سعد
الله تعالى سبعون ألف حجاب من نور وما تسمع نفس
شيئاً من حسن تلك الحجب إلا زهقت نفسها
قال أبو الشيخ في العظمة ذكر حجب ربنا تبارك
<\Matan>وتعالى فبدأ بهذا الحديث ثم بعده

Figure 6.6: An example of the confusion between an Isnad and Matan from the first experiment because of different Hadith structures in the training and testing sets.

We classified some Hadiths as hard Hadiths because of them having a story in the Isnad or between the Isnad and Matan, hence making the segmentation task more complex. There are two different types of these stories: a narrative story and chronology story. The narrative story refers to any story related to the narrator, such as describing where they lived, age, who they met and so on. A chronology story means telling a sequence of events in order (Sternberg 1990), such as describing the first event, which is the Prophet Muhammad and his companions' scene, why he said a certain Hadith or the person/group of people who came to ask him; the following event will be the Matan. We labelled the narrative story as Isnad and chronology story as Matan. Figure 6.7 shows an example of the narrative story wrongly predicted as a Matan. Figure 6.9 shows an example of the chronology story correctly predicted as a Matan from the second experiment.



<Isnad>وقال الطبراني حدثنا علي بن سعيد الرازي
حدثنا محمد بن حاتم المؤدب حدثنا القاسم بن
مالك المزني حدثنا سفيان بن زياد عن عمه سليم
بن زياد <\Isnad><Matan>قال لقيت عكرمة مولى
ابن عباس فقال لا تبرح حتى أشهدك على هذا الرجل
ابن لمعاذ بن عفراء فقال أخبرني بما أخبرك
أبوك عن قول رسول ﷲ صلى ﷲ عليه وسلم فقال
حدثني أبي أن رسول ﷲ صلى ﷲ عليه وسلم حدثه
أنه رأى رب العالمين عز وجل في حظيرة من القدس
في صورة شاب عليه تاج <\Matan>

Figure 6.7: : An example of the narrative story wrongly predicted as Matan using PPMD with order 7 (highlighted in blue).

6.3.2 Second Experiment

In this experiment, we used the Isnad and Matan training models that were produced from the first experiment. The LK Hadith corpus (Altammami et al. 2020) was chosen for testing purposes. It is a parallel corpus of English–Arabic Hadith, containing 39,038 annotated Hadiths from the six canonical Hadith books.

From the LK corpus, we manually extracted chapters two and three from the *Sahih Al-Bukhari* dataset, comprising a testing file of 10,539 words. We noticed that the last part of the Isnads, such as النبي صلى الله عليه وسلم قال '*the Prophet, may God bless him and grant him peace, said*', were labelled as Matans, so we relabelled these parts as Isnad for consistency with the labelling throughout. Then, we removed Arabic diacritics (Al-Tashkeel) and quotation marks.

Order 7 was chosen because it had a higher accuracy rate in the first experiment. The Hadith segmentation using PPMD produced an accuracy of 90.10%, an average precision of 0.92, an average recall of 0.86, and an average F-measure of 0.89. Figure 6.8 shows the confusion matrix of this experiment. Figure 6.9 shows an example of the chronology story correctly predicted as a Matan.

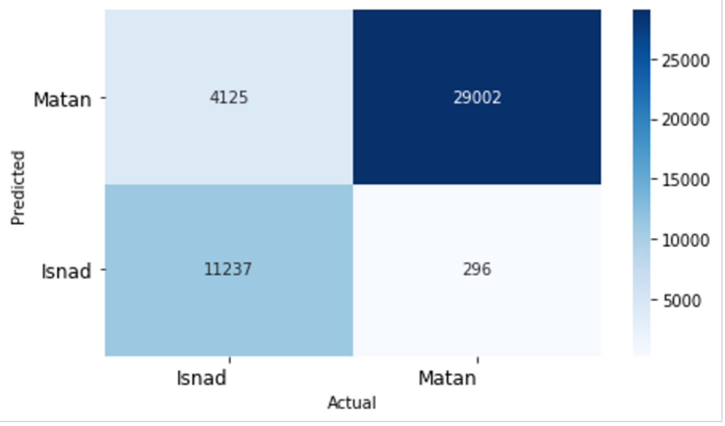
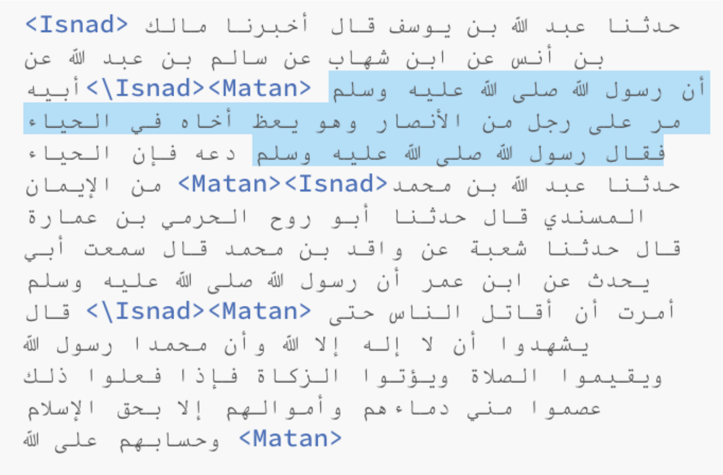


Figure 6.8: Confusion matrix of the second experiment's results.



حدثنا عبد الله بن يوسف قال أخبرنا مالك <Isnad>
بن أنس عن ابن شهاب عن سالم بن عبد الله عن
أن رسول الله صلى الله عليه وسلم <\Isnad><Matan>أبيه
مر على رجل من الأنصار وهو يعظ أخاه في الحياء
فقال رسول الله صلى الله عليه وسلم دعه فإن الحياء
حدثنا عبد الله بن محمد<Matan><Isnad> من الإيمان
المسندي قال حدثنا أبو روح الحرمي بن عمارة
قال حدثنا شعبة عن واقد بن محمد قال سمعت أبي
يحدث عن ابن عمر أن رسول الله صلى الله عليه وسلم
أمرت أن أقاتل الناس حتى <\Isnad><Matan> قال
يشهدوا أن لا إله إلا الله وأن محمدا رسول الله
ويقيموا الصلاة ويؤتوا الزكاة فإذا فعلوا ذلك
عصموا مني دماءهم وأموالهم إلا بحق الإسلام
وحسابهم على الله <Matan>

Figure 6.9: An example of the chronology story correctly predicted as a Matan from the second experiment (highlighted in blue).

6.4 Hadith Authenticity Classification

6.5 Goals for the Investigation

In this part, we first identify which part of the Hadith (Isnad, Matan or both) is the most effective for automatically detecting authenticity. Second, we examine the utilisation of DL and PPM classifiers, which have not previously been used for detecting Hadith authenticity. We then compare the DL, CML and PPM classifiers to determine which is the most effective classifier when detecting the authenticity of a Hadith.

6.6 Experimental Setup

Experiments in this part were a binary classification task for identifying the authentication for each Hadith (authentic or non-authentic). In this section, we describe the setup for the experiments. A supervised classification approach was adopted for Hadith authenticity classification by applying DL, CML and PPM algorithms using the NAH and LK corpora.

6.6.1 Dataset

The NAH corpus was used, which has been purposely designed for train and test non-authentic Hadith (see Chapter 3). The corpus covers a large number of NAH from non-famous Hadith books.

The LK corpus also was used to train and test authentic Hadith which is built by Altammami et al. (2020). The main advantage of this corpus is that it is freely available to the Hadith research community, while the main disadvantage is that the split into Isnad and Matan was automatically annotated and has only been manually verified for the *Sahih Al-Bukhari* subcorpus. This means that the other sub-corpora, such as *Sahih Muslim*, are noisy and need to be verified.

The experiments used two datasets ($N3\_1$ and $N3\_2$) from the NAH corpus and the *Sahih Al-Bukhari* and *Sahih Muslim* datasets from the LK corpus, here with the goal of determining authenticity. Each Hadith record was assigned to one of the following two classes:

- **Authentic** records that contained an authentic Hadith from the *Sahih Al-Bukhari* or *Sahih Muslim* datasets.
- **Non-authentic** records that contained NAH from the $N3\_1$ or $N3\_2$ datasets.

To guarantee a balanced distribution in the classes, we limited the number of records per class, thereby ensuring that the training phases used an equal number of records per category. In experiment A, we used 1,264 Hadith records from *Sahih Al-Bukhari* to train the authentic model, and we used 1,264 Hadith records from $N3\_1$ to train the non-authentic model. The testing file contained 2,996 Hadith records: 1,498 Hadith records from $N3\_2$ and 1,498 Hadith records from *Sahih Muslim*. In experiment B, the training file contained 2,996 Hadith records: 1,498 Hadith records from $N3\_2$ to train the non-authentic model and 1,498 Hadith records from *Sahih Muslim* to train the authentic model. The testing file contained 1,264 Hadith records from *Sahih Al-Bukhari* and 1,264 Hadith records from $N3\_1$.

We removed Hadith numbers, punctuation marks and non-Hadith text, and there is no further pre-processing was done, such as tokenisation, stemming or removal of stop words.

6.7 Experiments and Results

Three main experiments were performed to evaluate the automatic detection of Hadith authenticity: the PPM compression-based classifier; CML classifiers, such as the SVM, NB and DT classifiers ,implemented in Weka (Hall et al. 2009) with default parameters, and DL classifiers, including the LSTM, CNN and CNN–LSTM classifiers (Chapter 5).

These experiments were conducted to (1) detect Hadith authenticity based on Hadith; (2) detect Hadith authenticity based on Isnad; and (3) detect Hadith authenticity based on Matan. Aside from comparing the DL, CML and PPM classifiers, the primary aim of these experiments was to identify the part of a Hadith (Isnad, Matan or both) best used for effective automatic determination of authenticity. These experiments were novel in their approaches to Hadith authenticity classification because they investigated the use of the character-based text compression scheme PPM and DL classifiers.

6.7.1 Authentication based on Hadith

In the authentication based on Hadith experiments, there were two subexperiments (A and B), and each experiment had seven different experiments, in which each one applied a different classifier (see Figure 6.10).

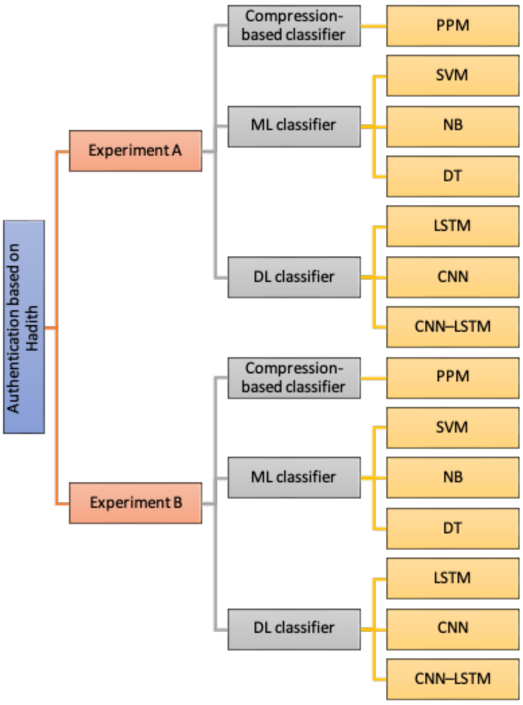


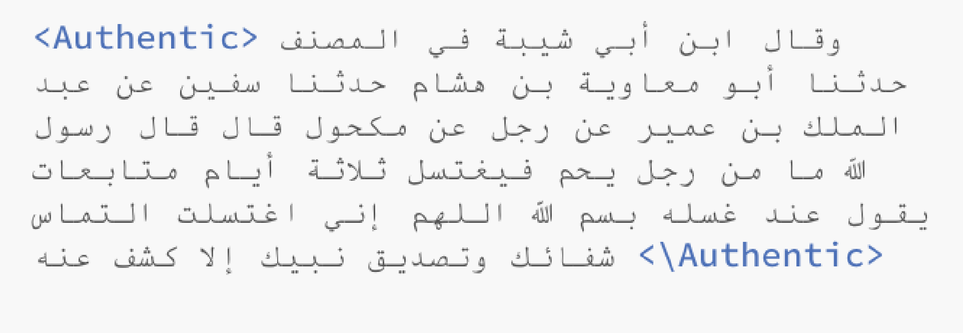
Figure 6.10: Authentication based on Hadith experiments.

Experiment A was a binary classification task for identifying the authentication for each Hadith (authentic or non-authentic). In experiment A, we extracted full Hadith records containing both Isnads and Matans from the *Sahih Al-Bukhari* and $N3\_1$ datasets to train the authentic and non-authentic models, respectively. The PPM and CNN-LSTM classifiers achieved higher rates of accuracy than the other classifiers, reaching up to 93%. The LSTM classifier obtained 80% and NB 76% accuracy. The lowest accuracy reported was from the DT classifier (55%). Table 6.3 shows the results of this experiment.

Table 6.3: The results of the authentication in the Hadith-based experiment, where B is Sahih Al-Bukhari and M is Sahih Muslim.

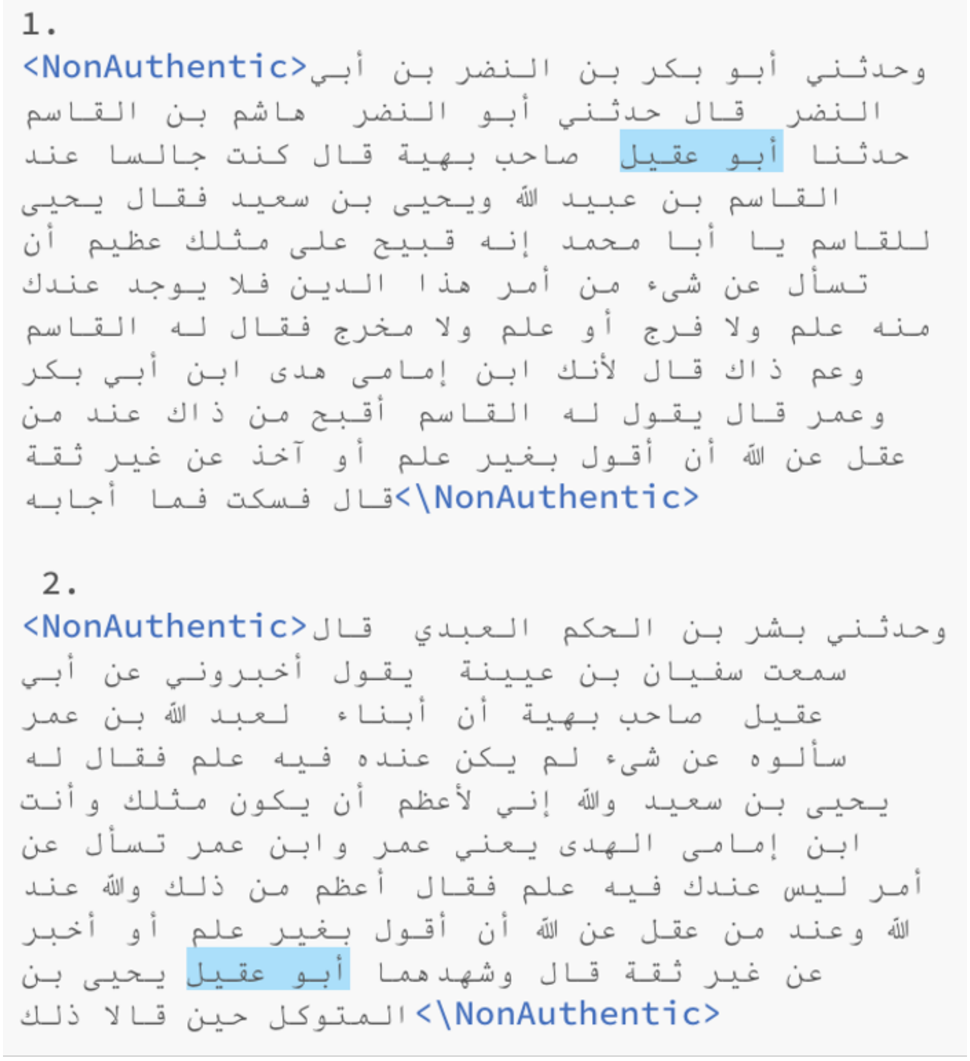
Experiment	Datasets	Classifier	Accuracy (%)	Recall	Precision	F-measure
A	Training sets: B/$N3\_1$ Testing sets: M/$N3\_2$	**PPM**	**93**	**0.94**	**0.93**	**0.93**
		SVM	61	0.61	0.78	0.54
		NB	76	0.76	0.83	0.75
		DT	55	0.55	0.76	0.44
		LSTM	80	0.97	0.73	0.84
		CNN	72	0.99	0.64	0.78
		CNN-LSTM	**93**	**0.93**	**0.93**	**0.93**
B	Training sets: M/$N3\_2$ Testing sets: B/$N3\_1$	**PPM**	**99**	**0.99**	**0.99**	**0.99**
		SVM	97	0.973	0.973	0.973
		NB	95	0.95	0.95	0.95
		DT	89	0.89	0.89	0.89
		LSTM	96	0.97	0.95	0.96
		CNN	97	0.96	0.99	0.97
		CNN-LSTM	97	0.97	0.97	0.97

Figure 6.11 illustrates an non-authentic Hadith predicted to be authentic in the PPM output of experiment A. Figure 6.12 illustrates some authentic Hadiths that were predicted to be non-authentic. This might be because the narrator أبوعقيل (highlighted in blue) had been mentioned several times in $N3\_1$, which was the non-authentic training set or it might be because these Hadiths did not mention Prophet Muhammad at the end of Isnad.



```
<Authentic> وقال ابن أبي شيبة في المصنف
حدثنا أبو معاوية بن هشام حدثنا سفين عن عبد
الملك بن عمير عن رجل عن مكحول قال قال رسول
ﷲ ما من رجل يحم فيغتسل ثلاثة أيام متابعات
يقول عند غسله بسم ﷲ اللهم إني اغتسلت التماس
شفائك وتصديق نبيك إلا كشف عنه <\Authentic>
```

Figure 6.11: Sample of an non-authentic Hadith predicted to be authentic in the PPM output of experiment A.



1.
<NonAuthentic>وحدثني أبو بكر بن النضر بن أبي
النضر قال حدثني أبو النضر هاشم بن القاسم
حدثنا أبو عقيل صاحب بهية قال كنت جالسا عند
القاسم بن عبيد الله ويحيى بن سعيد فقال يحيى
للقاسم يا أبا محمد إنه قبيح على مثلك عظيم أن
تسأل عن شيء من أمر هذا الدين فلا يوجد عندك
منه علم ولا فرج أو علم ولا مخرج فقال له القاسم
وعم ذاك قال لأنك ابن إمامي هدى ابن أبي بكر
وعمر قال يقول له القاسم أقبح من ذاك عند من
عقل عن الله أن أقول بغير علم أو آخذ عن غير ثقة
قال فسكت فما أجابه<\NonAuthentic>

2.
<NonAuthentic>وحدثني بشر بن الحكم العبدي قال
سمعت سفيان بن عيينة يقول أخبروني عن أبي
عقيل صاحب بهية أن أبناء لعبد الله بن عمر
سألوه عن شيء لم يكن عنده فيه علم فقال له
يحيى بن سعيد والله إني لأعظم أن يكون مثلك وأنت
ابن إمامي الهدى يعني عمر وابن عمر تسأل عن
أمر ليس عندك فيه علم فقال أعظم من ذلك والله عند
الله وعند من عقل عن الله أن أقول بغير علم أو أخبر
عن غير ثقة قال وشهدهما أبو عقيل يحيى بن
المتوكل حين قالا ذلك<\NonAuthentic>

Figure 6.12: Sample of authentic Hadiths predicted to be non-authentic in the PPM output of experiment A.

Experiment B was a binary classification task for identifying the authentication for each Hadith (authentic or non-authentic).In experiment B, we extracted full Hadith records containing both Isnads and Matans from the *Sahih Muslim* and *N*3_2 datasets to train the authentic and non-authentic models, respectively. This experiment yielded good results compared with the previous experiment. The best result was obtained using the PPM classifier, here with an accuracy of 99%. The SVM, CNN and CNN-LSTM classifiers obtained an accuracy of 97%. The accuracy of the DT classifier was the lowest of the other algorithms, with an accuracy of 89%. We believe that the enhancing in the experiment B classification result because of the large training size.

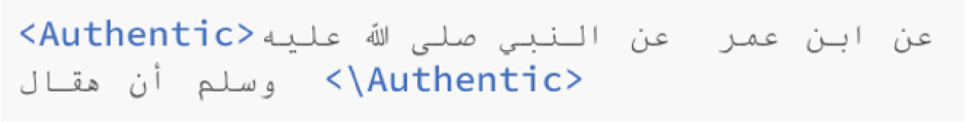
6.7.2 Authentication based on Isnad

Much like the authentication based on Hadith experiment, in this experiment, there were two subexperiments (A and B), and each experiment had seven different experiments, in which each one applied a different classifier. In experiment A, we extracted records that contained only Isnads from the *Sahih Al-Bukhari* and $N3\_1$ datasets to train the authentic and non-authentic models, respectively. The CNN classifier achieved better accuracy than the other classifiers and reached up to 93%. This was followed by the PPM classifier (92%), and then the SVM classifier (91%). The lowest accuracy was reported for the CNN–LSTM classifier (84%). Table 6.4 presents the results of this experiment.

Table 6.4: The results of the authentication based on Isnad experiment, where B is Sahih Al-Bukhari and M Sahih Muslim.

Experiment	Datasets	Classifier	Accuracy (%)	Recall	Precision	F-measure
A	Training sets: B/$N3\_1$ Testing sets: M/$N3\_2$	PPM	92	0.93	0.92	0.93
		SVM	91	0.91	0.92	0.92
		NB	89	0.89	0.90	0.89
		DT	90	0.90	0.90	0.90
		LSTM	90	0.95	0.87	0.90
		CNN	**93**	**0.97**	**0.90**	**0.93**
		CNN-LSTM	84	0.97	0.77	0.86
B	Training sets: M/$N3\_2$ Testing sets: B/$N3\_1$	**PPM**	**99**	**0.99**	**0.99**	**0.99**
		SVM	93	0.93	0.93	0.93
		NB	93	0.93	0.93	0.93
		DT	92	0.92	0.92	0.92
		LSTM	93	0.94	0.93	0.93
		CNN	96	0.97	0.96	0.96

Figure 6.13 illustrates an example of a non-authentic Isnad from $N3\_2$ that was predicted to be authentic. This is because the Isnad for this Hadith had a narrator, ابن عمر who is known to be a trustworthy narrator. It is possible that this Hadith was classified as authentic not because of any weakness in the Isnad but because of its Matan.



```
<Authentic>عن ابن عمر  عن النبي صلى الله عليه
وسلم أن هقال  <\Authentic>
```

Figure 6.13: Example of a non-authentic Isnad from $N3\_2$ predicted to be authentic in the PPM output of the second experiment (A).

In experiment B, we extracted records that contained only Isnads from the *Sahih Muslim* and

$N3\_2$ datasets to train the authentic and nonauthentic models, respectively. By comparing CML classifiers, the SVM and NB classifiers outperformed the DT classifier, here with an accuracy of 93%. On other hand, the CNN classifier outperformed the other DL classifiers, here with an accuracy of 96%. Overall, the PPM classifier outperformed all the other classifiers that were tested, here with an accuracy of 99%.

6.7.3 Authentication based on Matan

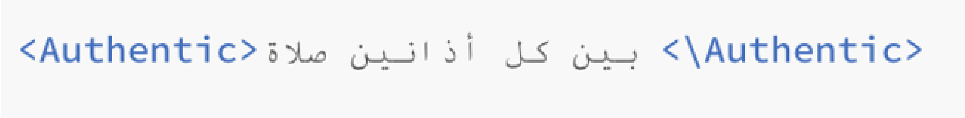
As the previous experiments, in this experiment, there were two sub-experiments (A and B), and each experiment had seven different experiments, in which each one applied a different classifier. In experiment A, we extracted Matan records, which contained only Matans, from the *Sahih Al-Bukhari* and $N3\_1$ datasets to train the authentic and the non-authentic models, respectively. The LSTM classifier achieved the highest rates of accuracy, reaching 85%, which was lower than the previous experiments. This was followed by the CNN and CNN–LSTM classifiers, at 84% and 82%, respectively. The PPM classifier obtained an accuracy of 79%. The lowest accuracy was reported by the SVM and DT classifiers, each reaching 55%. Table 6.5 provides the results of this experiment.

Table 6.5: The results of the authentication based on Matan experiment where B is *Sahih Al-Bukhari* and M is *Sahih Muslim*.

Experiment	Datasets	Classifier	Accuracy (%)	Recall	Precision	F-measure
A	Training sets: B/$N3\_1$ Testing sets:M/$N3\_2$	PPM	79	0.79	0.79	0.79
		SVM	55	0.56	0.75	0.45
		NB	72	0.72	0.80	0.70
		DT	55	0.56	0.75	0.45
		LSTM	**85**	**0.80**	**0.90**	**0.85**
		CNN	84	0.85	0.84	0.84
		CNN-LSTM	82	0.87	0.79	0.83
B	Training sets: M/$N3\_2$ Testing sets: B/$N3\_1$	PPM	82	0.85	0.82	0.84
		SVM	**88**	**0.88**	**0.90**	**0.88**
		NB	83	0.83	0.86	0.83
		DT	86	0.86	0.88	0.86
		LSTM	87	0.78	0.94	0.85
		CNN	**88**	**0.78**	**0.96**	**0.87**
		CNN-LSTM	**88**	**0.86**	**0.90**	**0.88**

Figure 6.14 illustrates an example of a non-authentic Matan from $N3\_2$ that was predicted as being authentic. This Matan was mentioned in the *Sahih Al-Bukhari* dataset several times. Furthermore, this Hadith might be narrated by different Isnads, and the Isnad mentioned in

the $N3\_2$ dataset constitutes a weakness.



<Authentic>بين كل أذانين صلاة<\Authentic>

Figure 6.14: Example of a non-authentic Matan from $N3\_2$ predicted to be authentic in the PPM output of the third experiments (A).

In experiment B, we extracted records that contained only Matans from the *Sahih Muslim* and $N3\_2$ datasets to train the authentic and non-authentic models, respectively. The SVM, CNN and CNN-LSTM achieved the best results, here with an accuracy of 88%. The accuracy of the PPM classifier was the lowest of the other algorithms, with an accuracy of 82%.

6.8 Discussion

By comparing A's experiments, First, the accuracy for the Authentication based on Hadith experiment ranged from 55% to 93%. The accuracy for the Authentication based on Isnad experiment was between 84% and 93%, while the accuracy for this Authentication based on Matan experiment ranged from 55% to 85%, meaning that the Isnad was the part of the Hadith that resulted in the most effective automatic determinations of authenticity. However, these experiments also proved that we could use the Matan to judge Hadiths, here with an accuracy rate of 85%.

Second, Figure 6.15 demonstrates how the PPM and the CNN-LSTM classifiers reported the best results compared with the other classifiers in the authentication based on Hadith experiment (blue bars). Also, authentication based on the Isnad reported the best results when using the CML classifiers, CNN and LSTM compared with the other classifiers (orange bars). In contrast, the LSTM classifier reported the best results compared with other classifiers in authentication based on Matans (grey bar).

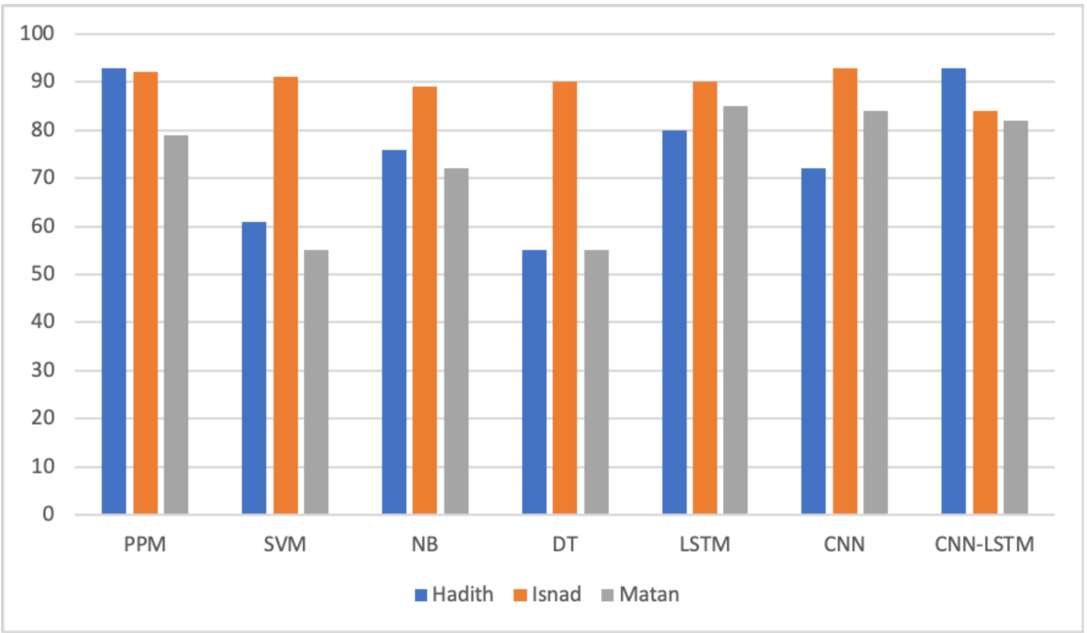


Figure 6.15: A comparison of the accuracy rates from A experiments using different parts of a Hadith.

By comparing B's experiments, the accuracy for the Authentication based on Hadith experiment ranged from 89% to 99%. The accuracy for the Authentication based on Isnad experiment was between 92% and 99%, while the accuracy for the Authentication based on Matan experiment ranged from 82% to 88%, meaning that the Isnad was the part of a Hadith that resulted in the most effective automatic determinations of authenticity. However, the B experiments also proved that we could use the Matan to judge Hadiths, here with an accuracy rate of 88% (Figure 6.16).

Figure 6.16 demonstrates how the PPM classifier reported the best results compared with the other classifiers in the authentication based on Hadith and authentication based on Isnad experiments (blue bars and orange bars, respectively). In contrast, the SVM and CNN–LSTM classifier reported the best results compared with the other classifiers in the authentication based on the Matan experiment (grey bar).

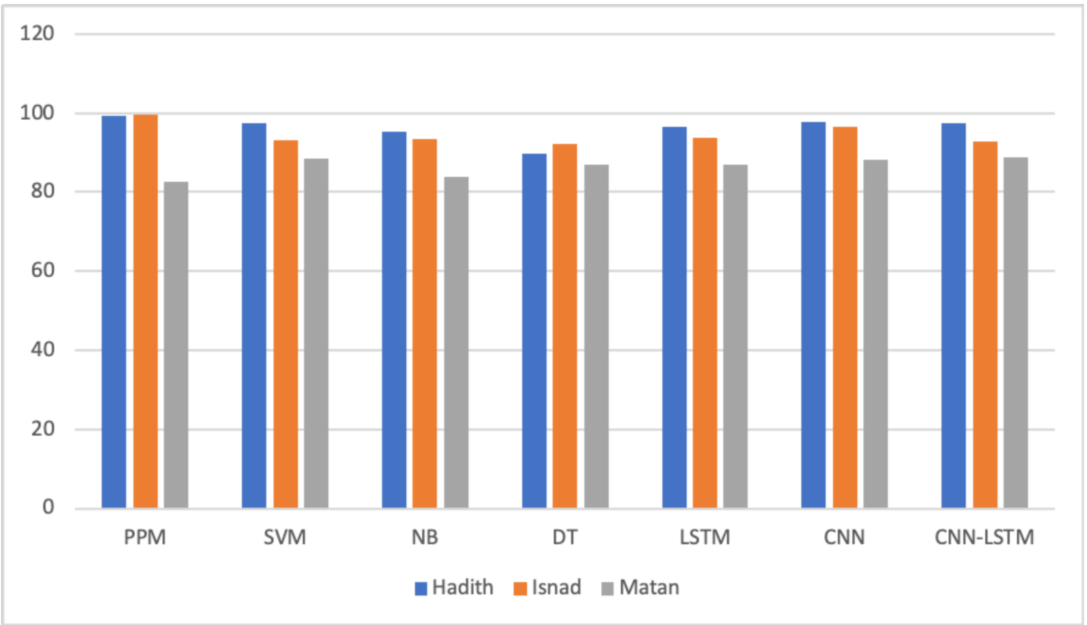


Figure 6.16: A comparison of the accuracy rates from the B experiments using the different parts of a Hadith.

By comparing the A and B experiments, we found out that the accuracy of the A experiments ranged from 55% and 93%. Although the accuracy of the B experiments ranged from 82% and 99%, we believe that this was because of the training set size. Hence, the performance of Hadith authenticity classification was significantly affected by the training size.

6.9 Conclusion and Future Work

This chapter aimed to answer the following research question: how does the effectiveness of the PPM compression-based approach compare against classical ML and DL algorithms for classifying Arabic Hadith text according to Hadith authenticity? First, we evaluated the PPM compression-based method for automatic segmentation of Arabic Hadith. The experiments showed that PPMD is effective in segmenting Hadiths into two main components (Isnad and Matan), here testing these on Hadith corpora (NAH and LK) that have different structures. The main innovation in these experiments is their use of a character-based text compression method to segment Hadiths.

For training the Isnad and Matan models, we used the first dataset in the NAH corpus. In the first experiment, we used the third dataset in the NAH corpus, which lacks a clear structure as a testing set. We found that the PPMD of order 7 obtained a higher accuracy (92.76%)

than other orders. In the second experiment, we aimed to evaluate PPMD segmentation on a different Hadith corpus, so we used the *Sahih Al-Bukhari* dataset of the LK Hadith corpus for testing purposes, which produced an accuracy of 90.10%.

The first experiment showed that the Hadith rank was not in the same place between the training and testing set, leading to some confusion between Isnad and Matan. Possible ways to reduce this confusion that could be undertaken in future work may be to (1) extend the Isnad training set to have different Isnads structured from different Hadith datasets or (2) clean the testing set of all non-Isnad words.

Second, this chapter has presented detailed research on the ways to automatically detect Hadith authenticity in Arabic Hadith texts. It has examined the utilisation of deep learning-based and PPM compression-based classifiers, which have not been previously used in detecting Hadith authenticity. The proposed methods were compared with the most recent method, that is, CML.

Our experiments showed that the Isnad is the part of a Hadith that results in the most effective automatic determination of authenticity. Also, the Matan can be used to judge Hadiths with an accuracy rate of up to 88%. Finally, we also demonstrated that the PPM and DL classifiers were the most effective means of automatically detecting authentic Hadiths. These results are very promising. However, we cannot really compare our results with the previous best result achieved by Aldhaln et al. (2012) and Najiyah et al. (2017) (discussed in section 2.8 in Chapter 2) because of the different testing texts used and these were not available.

From our experiments, duplicate Matans with different Isnads and authenticity labels created confusion in the classification results. As part of future work, a possible solution to overcome this issue would be to remove them from the training and testing sets to avoid training/testing the models on incoherent examples.

Also, future work could look at pretrained transformers, such as AraBERT. However, this may not work very well because it was trained on 70 million sentences from Arabic news text, which are entirely different from Hadith text in both vocabulary and spelling.

Chapter 7

Conclusion and Future Work

The work in the current thesis has been inspired by four motivations. The first was the limitation of available Arabic dialectal, Arabic code-switching and Arabic Hadith resources, even though there are a significant number of MSA resources in the Arabic NLP community. Also, the existing Arabic corpora are quite expensive and/or are of poor quality. Hence, the aim was to enrich Arabic resources by building several Arabic corpora and making them freely available to the Arabic research community.

Therefore, several Arabic corpora were built: the BAEC corpus, SDC, EDC and NAH corpus. These corpora were built to support various Arabic NLP research, such as detection of code-switching, dialect identification and Hadith classification. The BAEC corpus has 446,081 characters and 45,251 words and focuses on switching between Arabic and English. It includes code-switching between: MSA and English; the Saudi dialect and English; and the Egyptian dialect and English. Manually annotated, it was produced in XML. To the best of our knowledge, this corpus is the first freely available Arabic code-switching corpus derived from Facebook. A 210,396-word corpus called the SDC was built for training the Saudi model, and it contains the mixed dialects of Saudi Arabia. It was collected from social media platforms, such as Facebook and Twitter, and is 2,018 KB in size. The EDC, which we constructed for training the Egyptian model, consists of 218,149 words and is 2,024 KB in size. It was also collected from the social media platform Facebook. The SDC and EDC can be used as a reference for Arabic dialects. Finally, the NAH is a corpus containing Arabic Hadiths from lesser-known Hadith books, and it is a new resource for Hadith community research. It consists of 1,621,423 words from 15 non-famous Hadith books. To the best of our knowledge, this corpus is the first freely available

Arabic Hadith corpus derived from lesser-known Hadith books.

The second motivation was to detect code-switching in Arabic varieties and dialects from social media platforms to evaluate the PPM character-based approach and compare this against an SVM classifier with character-based approach and word-based approach. To the best of our knowledge, no previous study involving the detection of code-switching between Arabic and English using PPM compression has been published before. We investigated code-switching in Arabic Facebook text. This was done to (1) detect code-switching between the Egyptian dialect and English; (2) detect code-switching among the Egyptian dialect, the Saudi dialect and English; and (3) detect code-switching among the Egyptian dialect, the Saudi dialect, MSA and English.

The experimental results reported that PPM compression achieved a higher accuracy rate than SVM classifier when the training corpus correctly represented the language or dialect under study. When this condition has been satisfied, the compression-based approach will be more effective for automatically detecting code-switching in written Arabic text. Second, when Arabic and English are classified using SVM, the character filter is a more appropriate filter than the Word filter because the difference between these two languages is best modelled using characters. Third, the character filter is also more appropriate for a comparison between SVM and PPM because PPM is a character-based model.

The third motivation was to perform classifying Arabic Hadith experiments to evaluate the PPM compression approach and compare these against the classical ML and DL approaches. The aim was to classify Arabic Hadith into two main classification tasks: Hadith components classification and Hadith authenticity classification. In term of Hadith components classification, the experimental results showed that DL classifiers achieve a higher classification accuracy than the other classifiers. However, the execution time for the DL classifiers was high, even though they were using GPU hardware.

In term of Hadith authenticity classification, the experimental results showed that the Isnad is the part of a Hadith that results in the most effective automatic determination of authenticity. In addition, the Matan can be used to judge Hadiths, with an accuracy rate of up to 85%. Finally, the PPM and DL classifiers were effective means of automatically detecting authentic Hadiths. These results are very promising.

The fourth motivation was to evaluate a PPM compression-based method for the automatic segmentation of Arabic Hadith. The experiments showed that PPM can be effective in segmenting a Hadith into its two main components (Isnad and Matan); this was tested on different Hadith corpora that have different structures. The main innovation in these experiments was their use of a character-based text compression method to segment the Hadiths.

7.1 Review of Aim and Objectives

Section 1.1 has summarised the aim and objectives of this research. The PPM compression, CML algorithms and DL algorithms were successfully applied to Arabic text classification.

The specific objectives and how they have been achieved are as follows:

- *Create a corpus of Arabic Hadiths using Hadith websites.*

  This was achieved in Chapter 3 by presenting the creation of an NAH corpus that contains Arabic Hadith from lesser-known Hadith books, providing a new resource for Hadith researchers. It consists of 1,621,423 words from 15 non-famous Hadith books.

- *Create corpus of Arabic code-switching texts by using samples obtained from online sources.*

  This was achieved in Chapter 3 by presenting the creation of the BAEC corpus, which contains 446,081 characters and 45,251 words and focuses on switching between Arabic and English.

- *Create new corpora containing samples of text from different Arabic dialects such as Egyptian and Saudi.*

  This objective was also accomplished in Chapter 3 by presenting the creation of two Arabic dialect corpora: SDC and EDC. The SDC has 2,065,867 characters and 210,396 words and contains mixed dialects of Saudi Arabia from social media, such as Facebook and Twitter. The EDC consists of 2,072,165 characters, is around 218,149 words and contains an Egyptian dialect from Facebook.

- *Adapt the PPM compression-based approach in different applications (using the corpora that were created for previous the objectives) such as detecting code-switching in varieties and dialects, Hadith component categorisation, Hadith segmentation of Isnads and Matans and Hadith authenticity classification*

This was accomplished, as described in Chapters 4, 5, 6 by performing experiments to evaluate the PPM compression-based approach.

- *Compare the compression-based approach, CML classifiers and the DL classifiers to the automatic classification of Arabic Hadiths.*

  This was also accomplished, as described in Chapters 5and 6, by conducting experiments to compare the PPM compression-based approach, CML classifiers and DL classifiers to the automatic classification of Arabic Hadiths.

- *Compare the execution time of these methods to discover the faster and slower method of classifying Arabic Hadith.*

  This objective was achieved in Chapter 5 by performing experiments to compare the execution time of these methods to discover the faster and slower method of classifying Arabic Hadiths. The results showed that the DL classifiers produced classification accuracy better than the other classifiers, but the execution time was high, even though they were using GPU hardware. Hence, CML classifiers are faster.

The sub-objectives of current study have been achieved as follows:

- *Investigate the best order o of PPM for Arabic Hadith segmentation, where o is the number of characters used for predication.*

  This objective was accomplished in Chapter 6 by investigating which order was the best order for Arabic Hadith segmentation. We found that PPM of order 7 obtained a higher accuracy (92.76%) than the other orders.

- *Conduct a performance evaluation of the CML for text classification using different features such as word and character features.*

  This objective was achieved in Chapter 5 by investigating the performance evaluation of the CML for text classification using different features, such as word and character features.

- *Identify the part of a Hadith (Isnad, Matan or both) that is best used for effective automatic determination of authenticity.*

  This was also accomplished in Chapter 6. Our experiments showed that the Isnad is the part of a Hadith resulting in the most effective automatic determination of authenticity. The Matan can be used to judge Hadiths, with an accuracy rate of up to 85%.

- *Study the impact of training parameters such as training corpus size on the classification results.* This objective was achieved in Chapter 5 by performing experiments to study the impact of the training corpus size on the classification results.

7.2 Review of Research Question

The specific research questions underlying the current research are as follows:

- *How does the effectiveness of the PPM compression-based approach compare against traditional machine algorithms and DL algorithms for classifying Arabic Hadith text based on Hadith components and authenticity?*

  As shown in the experiments of Chapters 5 and 6, the PPM compression-based approach compared with traditional machine algorithms and DL algorithms was successfully applied to the problem of Arabic Hadith classification. The experimental results in Chapter 5 showed that DL classifiers achieved a higher classification accuracy than the other classifiers when classifying Hadiths according to their main components. On the other hand, the results in Chapter 6 showed that the PPM and DL classifiers were effective in automatically detecting authentic Hadiths. These results are very promising.

- *How does the effectiveness of the PPM compression-based approach compare against traditional machine algorithms for detecting code-switching in varieties and dialects from social media platforms?*

  In Chapter 4, we compared the PPM compression-based approach and CML classifier SVM to the automatic detection of code-switching in Arabic text. Our experiments showed that PPM achieved a higher accuracy rate than SVM when the training corpus correctly represents the language or dialect under study.

7.3 Limitations of the Work

While working on the current thesis, we have encountered a few limitations and because of time constraints, we were unable to verify them:

- When performing the detection of code-switching among the Egyptian dialect, the Saudi dialect, MSA and English, we found that the MSA corpus used to train the MSA PPM model in the third experiment did not represent the MSA text in Facebook because it was built from news websites. The most suitable solution to overcome this issue would be to build a new MSA Facebook corpus trained on MSA text specially taken from Facebook.
- The TEng and MAE words (discussed in Section 3.2.2 in Chapter 3) provided one of the biggest challenges for NLP tools because this issue was unforeseen and we had insufficient training data to provide an effective means to identify these phenomena.
- When applying DL classifiers using Tensorflow and keras, the classification error logs were difficult to analyse because everything was hidden.
- When automatically detecting Hadith authenticity in Arabic Hadith texts, duplicate Matans with different Isnads and authenticity labels created confusion in the classification results. A possible solution to overcome this issue was to remove them from the training and testing sets to avoid training/testing the models on incoherent examples.

7.4 Future Work

Future work can be as follows:

- Translation of code-switched tweets/posts into pure Arabic. This will help monolingual users to understand code-switched tweets/posts.
- Build a new MSA Facebook corpus to train on the MSA model and this will help Arabic NLP researchers who work on social media text.
- Further investigation is needed to study how the training corpus size affects the PPM classification results for Arabic text.
- DL approaches need further analysis to investigate the classification error logs.
- Complex DL models can be applied to determine whether they are better than simple DL models. However, this will require adding different CNN and LSTM layers.
- Classifying Hadiths by their topics can be applied using DL and PPM classifiers. This will help Hadith researchers who concentrate on Matan.

- Also, as part of future work, a pretrained transformer such as AraBERT can also be applied to Hadith text. However, this may not work very well because it was trained on 70 million sentences from Arabic news text, which are entirely different from Hadith text in both vocabulary and spelling.

References

Ababneh, Jafar, Almomani, Omar, Hadi, Wael, El-Omari, Nidhal Kamel Taha, and Al-Ibrahim, Ali (2014). “Vector space models to classify Arabic text”. In: *International Journal of Computer Trends and Technology (IJCTT)* 7.4, pp. 219–223.

Abbas, Qaiser, Ahmed, MS, and Niazi, Sadia (2010). “Language Identifier for Languages of Pakistan Including Arabic and Persian”. In: *International Journal of Computational Linguistics (IJCL)* 1.03, pp. 27–35. URL: https://www.researchgate.net/profile/Qaiser-Abbas-2/publication/230683559_Language_Identifier_for_Languages_of_Pakistan_Including_Arabic_and_Persian/links/09e41502ee9d93b886000000/Language-Identifier-for-Languages-of-Pakistan-Including-Arabic-and-Persian.pdf.

Ahmad, Gazi Imtiyaz and Singla, Jimmy (2022). “Machine learning approach towards language identification of Code-Mixed Hindi-English and Urdu-English Social Media Text”. In: *2022 International Mobile and Embedded Technology Conference (MECON)*, pp. 215–220. DOI: 10.1109/MECON53876.2022.9751958.

Ahmed, Arfan, Ali, Nashva, Alzubaidi, Mahmood, Zaghouani, Wajdi, Abd-alrazaq, Alaa A, and Househ, Mowafa (2022). “Freely Available Arabic Corpora: A Scoping Review”. In: *Computer Methods and Programs in Biomedicine Update* 2, p. 100049. ISSN: 2666-9900. DOI: https://doi.org/10.1016/j.cmpbup.2022.100049.

Al -Dashti, Abdulmohsen (1998). “Language choice in the state of Kuwait: A sociolinguistic investigation”. In: *Unpublished doctoral dissertation. University of Essex, Colchester, UK.*

Al -Kabi, Mohammed N, Wahsheh, Heider A, and Alsmadi, Izzat M (2014). “A topical classification of hadith Arabic text”. In: *IMAN* 2014, 2nd. URL: https://www.researchgate.net/

profile/Mohammed-Al-Kabi/publication/266796626_A_Topical_Classification_of_Hadith_Arabic_Text/links/543c33460cf2c432f7417124/A-Topical-Classification-of-Hadith-Arabic-Text.pdf.

Al -Moghrabi, Arwa Abdulrahman (2015). “An examination of reading strategies in Arabic (L1) and English (L2) used by Saudi female public high school adolescents”. PhD thesis. The British University in Dubai (BUiD). URL: https://bspace.buid.ac.ae/handle/1234/776.

Al -Shargabi, Bassam, Al-Romimah, Waseem, and Olayah, Fekry (2011). “A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination”. In: *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*. ISWSA ’11. Amman, Jordan: Association for Computing Machinery. ISBN: 9781450304740. DOI: 10.1145/1980822.1980833.

Al -Thubaity, Abdulmohsen O (2015). “A 700M+ Arabic corpus: KACST Arabic corpus design and construction”. In: *Springer* 49.3, pp. 721–751. URL: https://link.springer.com/article/10.1007/s10579-014-9284-1.

Alansary, Sameh and Nagi, Magdi (2014). “The international corpus of Arabic: Compilation, analysis and evaluation”. In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 8–17.

Aldhaln, Kawther A, Zeki, Akram M, and Zeki, Ahmed M (2012). “Knowledge extraction in hadith using data mining technique”. In: *International Journal of Information Technology and Computer Science* 2, pp. 13–21.

Alfaifi, Abdullah (2015). “Building the Arabic Learner Corpus and a System for Arabic Error Annotation”. PhD thesis. University of Leeds.

Alfaifi, Saeeda H (2013). *Code-switching among bilingual Saudis on Facebook*. Southern Illinois University at Carbondale.

Alhawiti, Khaled M (2014). *Adaptive models of Arabic text*. Bangor University (United Kingdom). URL: https://www.proquest.com/openview/7fbc5d890974cd0f5a644d0f4e4b611a/1?pq-origsite=gscholar&cbl=18750&diss=y.

Alkahtani, Saad (2015). *Building and verifying parallel corpora between Arabic and English.* Bangor University (United Kingdom). URL: https://www.proquest.com/openview/53460dcdb513d01385c5d3137553962b/1?pq-origsite=gscholar&cbl=51922.

Alkahtani, Saad and Teahan, William (2016). "A New Parallel Corpus of Arabic/English". In: *Proceedings of the Eighth Saudi Students Conference in the UK*, pp. 279–284. DOI: 10.1142/9781783269150_0024.

Alkhatib, Manar (2010). "Classification of Al-Hadith Al-Shareef using data mining algorithm". In: *European, Mediterranean and Middle Eastern Conference on Information Systems, EMCIS2010, Abu Dhabi, UAE*, pp. 1–23. URL: https://www.researchgate.net/profile/Manar-Alkhatib/publication/293100587_Classification_of_Al-Hadith_Al-Shareef_using_data_mining_algorithm/links/57fbc27d08ae329c3d497afe/Classification-of-Al-Hadith-Al-Shareef-using-data-mining-algorithm.pdf.

Alkhazi, Ibrahim and Teahan, William (2017). "Classifying and segmenting classical and modern standard Arabic using minimum cross-entropy". In: *International Journal of Advanced Computer Science and Applications* 8.4.

Almesfer, B. (2015). "An Exploration of Saudi-English code-switching in WhatsApp conversation". In: *Dissertation for MA TESOL. School of Education. University of Leicester.*

Alosaimy, Abdulrahman and Atwell, Eric (2017). "Sunnah Arabic Corpus: Design and Methodology." In: *Proceedings of the 5th International Conference on Islamic Applications in Computer Science and Technologies (IMAN 2017).* URL: https://eprints.whiterose.ac.uk/125569/.

Alrabiah, Maha, Al-Salman, AbdulMalik, and Atwell, Eric (2013). *The design and construction of the 50 million words KSUCCA.* University of Leeds. Reproduced with permission from the copyright holders. URL: https://eprints.whiterose.ac.uk/81860/.

Alshutayri, Areej and Atwell, Eric (2018). "Creating an Arabic dialect text corpus by exploring Twitter, Facebook, and online newspapers". In: *OSACT 3 Proceedings.* LREC. URL: https://eprints.whiterose.ac.uk/128607/.

Alshutayri, Areej, Atwell, Eric, Alosaimy, Abdulrahman, Dickins, James, Ingleby, Michael, and Watson, Janet (Dec. 2016). “Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts”. In: *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 204–211. URL: https://aclanthology.org/W16-4826.

Altamimi, Mohammed and Teahan, William (2017). “Gender and authorship categorisation of Arabic text from Twitter using PPM”. In: *International Journal of Computer Science and Information Technologies* 9, pp. 131–140.

Altammami, Shatha, Atwell, Eric, and Alsalka, Ammar (2019). “Text segmentation using n-grams to annotate Hadith corpus”. In: *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, pp. 31–39. URL: https://aclanthology.org/W19-5605.pdf.

– (2020). “The Arabic-English parallel corpus of authentic hadith”. In: *International Journal on Islamic Applications in Computer Science And Technology* 8.2. URL: https://www.researchgate.net/profile/Shatha-Altammami/publication/341359917_The_Arabic-English_Parallel_Corpus_of_Authentic_Hadith/links/5eecbdc0a6fdcc73be897794/The-Arabic-English-Parallel-Corpus-of-Authentic-Hadith.pdf.

Althobaiti, Maha J. (2021). “Creation of annotated country-level dialectal Arabic resources: An unsupervised approach”. In: *Natural Language Engineering*, pp. 1–42. DOI: 10.1017/S135132492100019X.

Alwedyan, Jaber, Hadi, Wa’el Musa, Salam, Ma’an, and Mansour, Hussein Y. (2011). “Categorize Arabic datasets using multi-Class classification based on association rule approach”. In: *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*. ISWSA ’11. Amman, Jordan: Association for Computing Machinery. ISBN: 9781450304740. DOI: 10.1145/1980822.1980840.

Amin, Muhammad Zain and Nadeem, Noman (2018). “Convolutional neural network: text classification model for open domain question answering system”. In: *arXiv preprint arXiv:1809.02479*.

Anderson, Benedict (2006). *Imagined communities: Reflections on the origin and spread of nationalism*. Verso Books.

Atwell, Eric (2018). “Classical and modern Arabic corpora: Genre and language change”. In: *Diachronic Corpora, Genre, and Language Change*. Ed. by RJ Whitt. Vol. 85. Studies in Corpus Linguistics. John Benjamins, pp. 65–91. DOI: 10.1075/scl.85.04atw.

– (2019). “Using the Web to model Modern and Quranic Arabic”. In: *Arabic Corpus Linguistics*. Ed. by T McEnery, A Hardie, and N Younis. Edinburgh, UK: Edinburgh University Press, pp. 100–119. URL: https://eprints.whiterose.ac.uk/131254/.

Azmi, Aqil and Bin Badia, Nawaf (2010). “iTree - automating the construction of the narration tree of Hadiths (Prophetic Traditions)”. In: *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010)*, pp. 1–7. DOI: 10.1109/NLPKE.2010.5587810.

Bahassine, Said, Madani, Abdellah, Al-Sarem, Mohammed, and Kissi, Mohamed (2020). “Feature selection using an improved Chi-square for Arabic text classification”. In: *Journal of King Saud University-Computer and Information Sciences* 32.2, pp. 225–231.

Bawaneh, Mohammed, Alkoffash, Mahmud, and Al Rabea, AI (2008). “Arabic text classification using K-NN and Naive Bayes.” In: *Journal of computer science* 4.4 (7), pp. 600–605.

Belinkov, Yonatan, Magidow, Alexander, Barrón-Cedeño, Alberto, Shmidman, Avi, and Romanov, Maxim (2018). “Studying the History of the Arabic Language: Language Technology and a Large-Scale Historical Corpus”. In: *CoRR* abs/1809.03891. arXiv: 1809.03891. URL: http://arxiv.org/abs/1809.03891.

Belinkov, Yonatan, Magidow, Alexander, Romanov, Maxim, Shmidman, Avi, and Koppel, Moshe (2016). “Shamela: a large-scale historical Arabic corpus”. In: *CoRR* abs/1612.08989. arXiv: 1612.08989. URL: http://arxiv.org/abs/1612.08989.

Benajiba, Yassine, Rosso, Paolo, and BenedíRuiz, José Miguel (2007). “ANERsys: an Arabic named entity recognition system based on maximum entropy”. In: *Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Berlin Heidelberg: Springer, pp. 143–153. ISBN: 978-3-540-70939-8.

Bilal, Kashif and Mohsin, Sajjad (2012). “Muhadith: a cloud based distributed expert system for classification of ahadith”. In: *2012 10th International Conference on Frontiers of Information Technology*, pp. 73–78. DOI: 10.1109/FIT.2012.22.

Boudad, Naaima, Faizi, Rdouan, Oulad Haj Thami, Rachid, and Chiheb, Raddouane (2018). “Sentiment analysis in Arabic: a review of the literature”. In: *Ain Shams Engineering Journal* 9.4, pp. 2479–2490. ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2017.04.007.

Boukil, Samir, Biniz, Mohamed, El Adnani, Fatiha, Cherrat, Loubna, and El Moutaouakkil, Abd Elmajid (2018). “Arabic text classification using deep learning technics”. In: *International Journal of Grid and Distributed Computing* 11.9, pp. 103–114. URL: https://www.researchgate.net/profile/Mohamed-Biniz/publication/327968032_Arabic_Text_Classification_Using_Deep_Learning_Technics/links/5ca73b9aa6fdcca26dff5dde/Arabic-Text-Classification-Using-Deep-Learning-Technics.pdf.

Brown, Jonathan AC (2009). *Hadith: Muhammad’s legacy in the medieval and modern world.* Simon and Schuster. URL: https://books.google.com.sa/books?id=0B69DwAAQBAJ&lpg=PT5&ots=FlbDDH_ICI&dq=BROWN%5C%2C%5C%20J.%5C%20A.%5C%20C.%5C%202009.%5C%20Hadith%5C%3A%5C%20Muhammad's%5C%20legacy%5C%20in%5C%20the%5C%20medieval%5C%20and%5C%20modern%5C%20world%5C%2C%5C%20ONEWORLD%5C%20CLASSICS.&lr&pg=PT5#v=onepage&q&f=false.

Brown, Peter F, Cocke, John, Della Pietra, Stephen A, Della Pietra, Vincent J, Jelinek, Frederick, Lafferty, John, Mercer, Robert L, and Roossin, Paul S (1990). “A statistical approach to machine translation”. In: *Computational linguistics* 16.2, pp. 79–85. URL: https://aclanthology.org/J90-2002.pdf.

Cavnar, William B and Trenkle, John M (1994). “N-gram-based text categorization”. In: *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval.* Vol. 161175. Citeseer. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.3248&rep=rep1&type=pdf.

Cleary, J. and Witten, I. (1984). “Data compression using adaptive coding and partial string matching”. In: *IEEE Transactions on Communications* 32.4, pp. 396–402. DOI: 10.1109/TCOM.1984.1096090.

Cohen, Jacob (1960). "A Coefficient of agreement for nominal sßcales". In: *Educational and Psychological Measurement* 20.1, pp. 37–46. DOI: 10.1177/001316446002000104.

Crystal, David (1987). "The Cambridge encyclopedia of language". In: *Cambridge University Press.*

Di Pietro, Robert (1977). "Code-switching as a verbal strategy among bilinguals". In: *Current themes in linguistics: Bilingualism, experimental linguistics and language typologies.*

Doyle, Charles (2016). *A dictionary of marketing.* Oxford University Press.

Duwairi, Rehab, Al-Refai, Mohammad Nayef, and Khasawneh, Natheer (2009). "Feature reduction techniques for Arabic text categorization". In: *Journal of the American society for information science and Technology* 60.11, pp. 2347–2352. DOI: https://doi.org/10.1002/asi.21173.

El -Haj, Mahmoud (2020). "Habibi-a multi dialect multi national Arabic song lyrics corpus". In: *European Language Resources Association (ELRA).* URL: https://eprints.lancs.ac.uk/id/eprint/142282/.

Eldin, Ahmad Abdel Tawwab Sharaf (2014). "Socio linguistic study of code switching of the Arabic language speakers on social networking". In: *International Journal of English Linguistics* 4.6, p. 78.

Elfardy, Heba, Al-Badrashiny, Mohamed, and Diab, Mona (2014). "Aida: identifying code switching in informal Arabic text". In: *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pp. 94–101. URL: https://aclanthology.org/W14-3911.pdf.

Elfardy, Heba and Diab, Mona (2012). "Token level identification of linguistic code switching". In: *Proceedings of COLING 2012: posters*, pp. 287–296. URL: https://aclanthology.org/C12-2029.pdf.

Elhassan, Rasha and Ahmed, Mahmoud (2015). "Arabic text classification review". In: *Evaluation* 12, p. 13.

Encoding Usage Distribution on the Entire Internet. (2022). Builtwith. URL: `https://trends.builtwith.com/encoding/traffic/Top-10k` (visited on 05/26/2022).

Evans, David (2007). "Corpus building and investigation for the Humanities". In: *University of Nottingham http://www. corpus. bham. ac. uk/corpus-building. shtml.*

Faidi, Kaouther, Ayed, Raja, Bounhas, Ibrahim, and Elayeb, Bilel (2014). "Comparing Arabic NLP tools for Hadith classification". In: *Proceedings of the 2nd International Conference on Islamic Applications in Computer Science and Technologies (IMAN'14).*

Farghaly, Ali and Shaalan, Khaled (Dec. 2009). "Arabic natural language processing: challenges and solutions". In: *ACM Transactions on Asian Language Information Processing* 8.4. ISSN: 1530-0226. DOI: `10.1145/1644879.1644881`.

Francis, W. N. and Kucera, H. (1979). *Brown corpus manual.* Tech. rep. Department of Linguistics, Brown University, Providence, Rhode Island, US. URL: `http://icame.uib.no/brown/bcm.html`.

Gal, Susan (1988). "The political economy of code choice". In: *Codeswitching: Anthropological and sociolinguistic perspectives* 48, pp. 245–64.

Gharib, Tarek F, Habib, Mena B, and Fayed, Zaki Taha (2009). "Arabic text classification using support vector machines." In: *International Journal of Computing.* 16.4, pp. 192–199.

Ghazizadeh, M., Zahedi, M.H., Kahani, M., and Bidgoli, B. Minaei (2008). "Fuzzy expert system in determining Hadith1 validity". In: *Advances in Computer and Information Sciences and Engineering.* Ed. by Tarek Sobh. Dordrecht: Springer Netherlands, pp. 354–359. ISBN: 978-1-4020-8741-7.

Goldberg, Yoav (2017). "Neural network methods for natural language processing". In: *Synthesis Lectures on human language technologies* 10.1, pp. 1–309. DOI: `doi.org/10.2200/S00762ED1V01Y201703HLT037`.

Graddol, David, Leith, Dick, and Swann, Joan (1996). *English: history, diversity, and change.* Vol. 1. Psychology Press. URL: `https://books.google.com.sa/books?id=nr6BhQAciWAC&lpg=PA1&ots=qYuFORy_vF&dq=GRADDOL%5C%2C%5C%20D.%5C%2C%5C%20LEITH%5C%2C%`

5C%20D.%5C%20%5C%26%5C%20SWANN%5C%2C%5C%20J.%5C%201996.%5C%20English%5C%3A%5C%20history%5C%2C%5C%20diversity%5C%2C%5C%20and%5C%20change%5C%2C%5C%20Psychology%5C%20Press.&lr&pg=PA1#v=onepage&q=GRADDOL,%5C%20D.,%5C%20LEITH,%5C%20D.%5C%20&%5C%20SWANN,%5C%20J.%5C%201996.%5C%20English:%5C%20history,%5C%20diversity,%5C%20and%5C%20change,%5C%20Psychology%5C%20Press.&f=false.

Grosjean, François (1982). *Life with two languages: An introduction to bilingualism.* Harvard University Press. URL: https://books.google.com.sa/books?id=VqGpxZ9pDRgC&lpg=PA1&ots=ARsgfJyg5j&dq=GROSJEAN%5C%2C%5C%20F.%5C%201982.%5C%20Life%5C%20with%5C%20two%5C%20languages%5C%3A%5C%20An%5C%20introduction%5C%20to%5C%20bilingualism%5C%2C%5C%20Harvard%5C%20University%5C%20Press.&lr&pg=PA1#v=onepage&q=GROSJEAN,%5C%20F.%5C%201982.%5C%20Life%5C%20with%5C%20two%5C%20languages:%5C%20An%5C%20introduction%5C%20to%5C%20bilingualism,%5C%20Harvard%5C%20University%5C%20Press.&f=false.

Gumperz, John J (1982). *Discourse strategies.* 1. Cambridge University Press. URL: https://books.google.com.sa/books?id=aUJNgHWl_koC&lpg=PR7&ots=jDCYTOO4Xg&dq=GUMPERZ%5C%2C%5C%20J.%5C%20J.%5C%201982.%5C%20Discourse%5C%20strategies%5C%2C%5C%20Cambridge%5C%20University%5C%20Press.&lr&pg=PR7#v=onepage&q=GUMPERZ,%5C%20J.%5C%20J.%5C%201982.%5C%20Discourse%5C%20strategies,%5C%20Cambridge%5C%20University%5C%20Press.&f=false.

Hakak, Saqib, Kamsin, Amirrudin, Zada Khan, Wazir, Zakari, Abubakar, Imran, Muhammad, Ahmad, Khadher bin, and Amin Gilkar, Gulshan (2020). "Digital Hadith authentication: recent advances, open challenges, and future directions". In: *Transactions on Emerging Telecommunications Technologies* n/a.n/a, e3977. DOI: https://doi.org/10.1002/ett.3977.

Hale, Scott A. (2014). "Global Connectivity and Multilinguals in the Twitter Network". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, pp. 833–842. ISBN: 9781450324731. DOI: 10.1145/2556288.2557203.

Hall, Mark, Frank, Eibe, Holmes, Geoffrey, Pfahringer, Bernhard, Reutemann, Peter, and Witten, Ian H. (Nov. 2009). "The WEKA data mining software: an update". In: *Special Interest*

Group on Knowledge Discovery in Data (SIGKDD) Explor. Newsl. 11.1, pp. 10–18. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278.

Harrag, Fouzi (2014). "Text mining approach for knowledge extraction in Sahîh Al-Bukhari". In: *Computers in Human Behavior* 30, pp. 558–566. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2013.06.035.

Harrat, Salima, Meftouh, Karima, Abbas, Mourad, Jamoussi, Salma, Saad, Motaz, and Smaili, Kamel (2015). "Cross-dialectal Arabic processing". In: *Computational Linguistics and Intelligent Text Processing.* Ed. by Alexander Gelbukh. Cham: Springer International Publishing, pp. 620–632. ISBN: 978-3-319-18111-0.

Hidayat, Taofik (2012). "An analysis of code-switching used by Facebookers (a case study in a social network site)". In: *Student essay for the study programme Pendidikan Bahasa Inggris (English Education) at STKIP Siliwangi Bandung, Indonesia.*

Hmeidi, Ismail, Hawashin, Bilal, and El-Qawasmeh, Eyas (2008). "Performance of KNN and SVM classifiers on full word Arabic articles". In: *Advanced Engineering Informatics* 22.1, pp. 106–111. ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2007.12.001.

Howard, Paul Glor (1993). "The design and analysis of efficient lossless data compression systems". PhD thesis. Brown University.

Hussein, Riyad F. (1999). "Code-alteration among Arab college students". In: *World Englishes* 18.2, pp. 281–289. DOI: https://doi.org/10.1111/1467-971X.00141.

Ibn Al-Salah, Al-Shahrazuri (1236). *Muqaddimah Ibn al-Salah 'Introduction to the Science of Hadith'.* Dar al-Ma'arif, Cairo, pp. 193–195.

Jogin, Manjunath, Mohana, Madhulika, M S, Divya, G D, Meghana, R K, and Apoorva, S (2018). "Feature extraction using convolution neural networks (CNN) and deep learning". In: *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 2319–2323. DOI: 10.1109/RTEICT42901.2018.9012507.

Johnson, Ian (2013). “Audience design and communication accommodation theory: use of Twitter by Welsh-English biliterates”. In: *Social Media and minority languages: Convergence and the creative industries*, pp. 99–118.

Jurafsky, Daniel and Martin, James H (2008). “Speech and language processing: an introduction to speech recognition, computational linguistics and natural language processing”. In: *Prentice Hall*.

Jurgens, David, Dimitrov, Stefan, and Ruths, Derek (2014). “Twitter users# codeswitch hashtags!# moltoimportante# wow”. In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pp. 51–61. URL: https://aclanthology.org/W14-3906.pdf.

Kennedy, Graeme (2014). *An introduction to corpus linguistics*. Routledge. DOI: https://doi.org/10.4324/9781315843674.

Khorsheed, Mohammad S and Al-Thubaity, Abdulmohsen O (2013). “Comparative evaluation of text classification techniques using a large diverse Arabic dataset”. In: *Language Resources and Evaluation* 47.2, pp. 513–538. URL: https://link.springer.com/article/10.1007/s10579-013-9221-8.

Khreisat, Laila (2006). “Arabic text classification using n-gram frequency statistics a comparative study.” In: *The 2006 International Conference on Data Mining (DMIN)*, pp. 78–82. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.3007&rep=rep1&type=pdf.

– (2009). “A machine learning approach for Arabic text classification using n-gram frequency statistics”. In: *Journal of Informetrics* 3.1, pp. 72–77. ISSN: 1751-1577. DOI: https://doi.org/10.1016/j.joi.2008.11.005.

Kilgarriff, Adam and Grefenstette, Gregory (Sept. 2003). “Introduction to the special issue on the web as corpus”. In: *Computational Linguistics* 29.3, pp. 333–347. ISSN: 0891-2017. DOI: 10.1162/089120103322711569.

Kowsari, Kamran, Brown, Donald E., Heidarysafa, Mojtaba, Jafari Meimandi, Kiana, Gerber, Matthew S., and Barnes, Laura E. (2017). “HDLTex: hierarchical deep learning for text classification". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364–371. DOI: 10.1109/ICMLA.2017.0-134.

Landis, J Richard and Koch, Gary G (1977). "The measurement of observer agreement for categorical data". In: *Biometrics*, pp. 159–174. DOI: https://doi.org/10.2307/2529310.

Li, Xuewei and Ning, Hongyun (2020). "Chinese text classification based on hybrid model of CNN and LSTM". In: *Proceedings of the 3rd International Conference on Data Science and Information Technology*, pp. 129–134. ISBN: 9781450376044. DOI: 10.1145/3414274.3414493.

Li, Yaoyong, Bontcheva, Kalina, and Cunningham, Hamish (2009). "Adapting SVM for data sparseness and imbalance: a case study in information extraction". In: *Natural Language Engineering* 15.2, pp. 241–271. DOI: 10.1017/S1351324908004968. URL: https://www.cambridge.org/core/journals/natural-language-engineering/article/abs/adapting-svm-for-data-sparseness-and-imbalance-a-case-study-in-information-extraction/3E3CDFD4C9B75A68A07A1314D2A346A5.

Lignos, Constantine and Marcus, Mitch (2013). "Toward web-scale analysis of codeswitching". In: *87th Annual Meeting of the Linguistic Society of America*. Vol. 90.

Lüdeling, Anke and Kytö, Merja (2008). *Corpus Linguistics: an international handbook, Volume 1*. De Gruyter Mouton. ISBN: 9783110211429. DOI: doi:10.1515/booksetHSK29.

MacWhinney, Brian (2000). *The CHILDES project: the database*. Vol. 2. Psychology Press. URL: https://books.google.com.sa/books?id=zxN648YXqHYC&lpg=PA1&ots=YQoQ9k6JEP&dq=MACWHINNEY%5C%2C%5C%20B.%5C%202000.%5C%20The%5C%20CHILDES%5C%20project%5C%3A%5C%20The%5C%20database%5C%2C%5C%20Psychology%5C%20Press.&lr&pg=PA1#v=onepage&q=MACWHINNEY,%5C%20B.%5C%202000.%5C%20The%5C%20CHILDES%5C%20project:%5C%20The%5C%20database,%5C%20Psychology%5C%20Press.&f=false.

Mahmood, Ahsan, Khan, Hikmat Ullah, Alarfaj, Fawaz K, Ramzan, Muhammad, and Ilyas, Mahwish (2018). "A multilingual datasets repository of the Hadith content". In: *International Journal of Advanced Computer Science and Applications* 9.2. URL: https://pdfs.semanticscholar.org/cc7a/e519a8cfbc89081a421a1afbf35881b6440b.pdf.

Malik, Lalita (1994). *Socio-linguistics: a study of code-switching.* Anmol Publications PVT. LTD.

Maraoui, Hajer, Haddar, Kais, and Romary, Laurent (2018). "Segmentation tool for Hadith corpus to generate TEI encoding". In: *International Conference on Advanced Intelligent Systems and Informatics*. Springer, pp. 252–260. URL: `https://link.springer.com/chapter/10.1007/978-3-319-99010-1_23`.

Mesleh, AM (2008). "Support vector machine text classifier for Arabic articles: Ant Colony optimization-based feature subset selection". In: *The Arab Academy for Banking and Financial Sciences.*

Milroy, James et al. (1995). *One speaker, two languages: cross-disciplinary perspectives on code-switching.* Vol. 10. Cambridge University Press. URL: `https://books.google.com.sa/books?id=7UV9Fel7A0YC&lpg=PR9&ots=R-Ob541zCP&dq=MILROY%5C%2C%5C%20J.%5C%201995.%5C%20One%5C%20speaker%5C%2C%5C%20two%5C%20languages%5C%3A%5C%20Cross-disciplinary%5C%20perspectives%5C%20on%5C%20code-switching%5C%2C%5C%20Cambridge%5C%20University%5C%20Press.&lr&pg=PR9#v=onepage&q=MILROY,%5C%20J.%5C%201995.%5C%20One%5C%20speaker,%5C%20two%5C%20languages:%5C%20Cross-disciplinary%5C%20perspectives%5C%20on%5C%20code-switching,%5C%20Cambridge%5C%20University%5C%20Press.&f=false`.

Mitchell, Tom M (1997). *Machine learning.* Vol. 1. 9. McGraw-hill New York.

Moffat, A. (1990). "Implementing the PPM data compression scheme". In: *IEEE Transactions on Communications* 38.11, pp. 1917–1921. DOI: `10.1109/26.61469`.

Muaad, Abdullah Y, Kumar, G Hemantha, Hanumanthappa, J, Benifa, JV Bibal, Mourya, M Naveen, Chola, Channabasava, Pramodha, M, and Bhairava, R (2022). "An effective approach for Arabic document classification using machine learning". In: *Global Transitions Proceedings* 3.1, pp. 267–271.

Myers-Scotton, Carol (2005). *Multiple voices: an introduction to bilingualism.* John Wiley & Sons.

Najeeb, Moath Mustafa Ahmad (2021). "Towards a deep leaning-based approach for Hadith classification". In: *European Journal of Engineering and Technology Research* 6.3, pp. 9–15. URL: https://www.ej-eng.org/index.php/ejeng/article/view/2378.

Naji Al-Kabi, Mohammed, Kanaan, Ghassan, Al-Shalabi, Riyad, Al-Sinjilawi, Saja I, and Al-Mustafa, Ronza S (2005). "Al-Hadith text classifier". In: *Journal of Applied Sciences* 5.3, pp. 584–587.

Najib, Mohammad, Abd Rahman, Nurazzah, Alias, N, Alias, MN, et al. (2017). "Comparative study of machine learning approach on Malay translated Hadith text classification based on Sanad". In: *MATEC Web of Conferences*. EDP Sciences. DOI: 10.1051/matecconf/201713500066.

Najiyah, Ina, Susanti, Sari, Riana, Dwiza, and Wahyudi, Mochammad (2017). "Hadith degree classification for Shahih Hadith identification web based". In: *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–6. DOI: 10.1109/CITSM.2017.8089304.

Nguyen, Dong, Doğruöz, A. Seza, Rosé, Carolyn P., and Jong, Franciska de (Sept. 2016). "Computational Sociolinguistics: A Survey". In: *Computational Linguistics* 42.3, pp. 537–593. ISSN: 0891-2017. DOI: 10.1162/COLI_a_00258.

Nwesri, Abdusalam FA, Tahaghoghi, Seyed MM, and Scholer, Falk (2005). "Stemming Arabic conjunctions and prepositions". In: *International symposium on string processing and information retrieval*. Springer, pp. 206–217.

Pasha, Arfath, Al-Badrashiny, Mohamed, Diab, Mona, El Kholy, Ahmed, Eskander, Ramy, Habash, Nizar, Pooleery, Manoj, Rambow, Owen, and Roth, Ryan (May 2014). "MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of Arabic". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1094–1101. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/593_Paper.pdf.

Peng, Nanyun, Wang, Yiming, and Dredze, Mark (2014). "Learning polylingual topic models from code-switched social media documents". In: *Proceedings of the 52nd Annual Meeting of* *the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 674–679. URL: https://aclanthology.org/P14-2110.pdf.

Platt, John (Apr. 1998). *Sequential minimal optimization: a fast algorithm for training support vector machines*. Tech. rep. MSR-TR-98-14. Microsoft. URL: https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.

Rosenfeld, R. (2000). "Two decades of statistical language modeling: where do we go from here?" In: *Proceedings of the IEEE* 88.8, pp. 1270–1278. DOI: 10.1109/5.880083.

Ryan, Matthew S and Nudd, Graham R (1993). "The viterbi algorithm". In: URL: https://wrap.warwick.ac.uk/60926/.

Saloot, Mohammad Arshi, Idris, Norisma, Mahmud, Rohana, Ja'afar, Salinah, Thorleuchter, Dirk, and Gani, Abdullah (2016). "Hadith data mining and classification: a comparative analysis". In: *Artificial Intelligence Review* 46.1, pp. 113–128. DOI: doi.org/10.1007/s10462-016-9458-x.

Sammut, Claude and Webb, Geoffrey I (2017). *Encyclopedia of machine learning and data mining*. Springer.

Saudi Arabia Social Media Statistics 2018 – Official GMI Blog (2019). Global Media Insight. URL: https://www.globalmediain-sight.com/blog/saudi-arabia-social-media-statistics/.

Siddiqui, Muazzam Ahmed, Saleh, ME, and Bagais, Abobakr Ahmed (2014). "Extraction and visualization of the chain of narrators from Hadiths using named entity recognition and classification". In: *International Journal of Computational Linguistics Research* 5.1, pp. 14–25.

Simons, Gary F and Fennig, Charles D (2017). *Ethnologue: languages of the world*. Vol. 12. 12. Dallas, Texas: SIL International.

Solorio, Thamar, Blair, Elizabeth, Maharjan, Suraj, Bethard, Steven, Diab, Mona, Ghoneim, Mahmoud, Hawwari, Abdelati, AlGhamdi, Fahad, Hirschberg, Julia, Chang, Alison, et al.

(2014). "Overview for the first shared task on language identification in code-switched data". In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pp. 62–72. URL: https://aclanthology.org/W14-3907.pdf.

Solorio, Thamar and Liu, Yang (2008). "Part-of-speech tagging for English-Spanish code-switched text". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1060. URL: https://aclanthology.org/D08-1110.pdf.

Sternberg, Meir (1990). "Telling in time (I): chronology and narrative theory". In: *Poetics Today* 11.4, pp. 901–948. DOI: doi.org/10.2307/1773082.

Swain, Philip H. and Hauska, Hans (1977). "The decision tree classifier: design and potential". In: *IEEE Transactions on Geoscience Electronics* 15.3, pp. 142–147. DOI: 10.1109/TGE.1977.6498972.

Syiam, Mostafa M, Fayed, Zaki T, and Habib, Mena B (2006). "An intelligent system for Arabic text categorization". In: *International Journal of Intelligent Computing and Information Sciences* 6.1, pp. 1–19. URL: https://ris.utwente.nl/ws/portalfiles/portal/6540474/IJICIS2006.pdf.

Teahan, William (2000). "Text classification and segmentation using minimum cross-entropy". In: *Content-Based Multimedia Information Access - Volume 2*. RIAO '00. Paris, France: Le Centre De Hautes Etudes Internationales D'Informatique Documentaire, pp. 943–961. URL: https://dl.acm.org/doi/abs/10.5555/2856151.2856154.

– (2018). "A compression-based toolkit for modelling and processing natural language text". In: *Information* 9.12. ISSN: 2078-2489. DOI: 10.3390/info9120294.

Trudgill, Peter (1983). *On dialect: social and geographical perspectives*. Wiley-Blackwell.

Viterbi, A. (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2, pp. 260–269. DOI: 10.1109/TIT.1967.1054010.

Warschauer, Mark, Said, Ghada R. El, and Zohry, Ayman G. (July 2002). “Language choice online: globalization and identity in Egypt”. In: *Journal of Computer-Mediated Communication* 7.4. ISSN: 1083-6101. DOI: `https://doi.org/10.1111/j.1083-6101.2002.tb00157.x`.

Wu, Chung-Hsien, Liu, Chao-Hong, Harris, Matthew, and Yu, Liang-Chih (2010). “Sentence correction incorporating relative position and parse template language models”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6, pp. 1170–1181. DOI: `10.1109/TASL.2009.2031237`.

Yu, Liang-Chih, He, Wei-Cheng, and Chien, Wei-Nan (2012). “A language modeling approach to identifying code-switched sentences and words”. In: *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 3–8. URL: `https://aclanthology.org/W12-6303.pdf`.

Yu, Liang-Chih, He, Wei-Cheng, Chien, Wei-Nan, and Tseng, Yuen-Hsien (2013). “Identification of code-switched sentences and words using language modeling approaches”. In: *Mathematical Problems in Engineering* 2013. DOI: `doi.org/10.1155/2013/898714`.

Yu, Liang-Chih, Wu, Chung-Hsien, Chang, Ru-Yng, Liu, Chao-Hong, and Hovy, Eduard (2010). “Annotation and verification of sense pools in OntoNotes”. In: *Information Processing Management* 46.4. Semantic Annotations in Information Retrieval, pp. 436–447. ISSN: 0306-4573. DOI: `https://doi.org/10.1016/j.ipm.2009.11.002`.

Zhang, Jiarui, Li, Yingxiang, Tian, Juan, and Li, Tongyan (2018). “LSTM-CNN hybrid model for text classification”. In: *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 1675–1680. DOI: `10.1109/IAEAC.2018.8577620`.

Zhang, Xiang, Zhao, Junbo, and LeCun, Yann (2015). “Character-level convolutional networks for text classification”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc.