# Classifying and Segmenting Classical and Modern Standard Arabic using Minimum Cross-Entropy

Ibrahim S Alkhazi
College of Computers & Information Technology
Tabuk University
Tabuk, Saudi Arabia

William J. Teahan
School of Computer Science Bangor University
United Kingdom

***Abstract*—Text classification is the process of assigning a text or a document to various predefined classes or categories to reflect their contents. With the rapid growth of Arabic text on the Web, studies that address the problems of classification and segmentation of the Arabic language are limited compared to other languages, most of which implement word-based and feature extraction algorithms. This paper adopts a PPM character-based compression scheme to classify and segment Classical Arabic (CA) and Modern Standard Arabic (MSA) texts. An initial experiment using the PPM classification method on samples of text resulted in an accuracy of 95.5%, an average precision of 0.958, an average recall of 0.955 and an average F-measure of 0.954, using the concept of minimum cross-entropy. PPM-based classification experiments on standard Arabic corpora showed that they contained different types of text (CA or MSA), or a mixture of the both (CA and MSA). Further experiments with the same corpora showed that a more accurate picture of the contents of the corpora was possible using the PPM-based segmentation method. Tag-based compression experiments (using tags produced by parts-of-speech Arabic taggers) also showed that the quality of the tagging (as measured by compression quality) is significantly affected when tagging either CA and MSA text. The conclusion is that NLP applications (such as taggers) should treat these texts separately and use different training data for each or process them differently.**

***Keywords*—text classification; Arabic language; Classical Arabic; Modern Standard Arabic***

## I. Introduction

Text classification is the process of automatically assigning a document to different predefined classes or categories to reflect their contents [1]. Text classification is important in various areas such as natural language processing (NLP), text mining, information retrieval, machine learning, etc. [2]. It also can be applied in a large variety of applications such as spam filtering [3], author identification [4]–[6], gender identification [7], [8], sentiment analysis [8]–[11], dialects identification [12], [13], and so on.

The massive increase in the size of text accessible on the internet during the last two decades has drawn the attention to the importance of text classification [2]. This increase of data on the Web has produced the need for methods to extract the required information from text documents, and therefore, generating unique difficulties for the text classification problem especially when considering applications requiring analysis of big data [2], [14].

Text classification can be implemented using various algorithms, for example, Naïve Bayes and the chain augmented Naïve Bayes probabilistic classifier [15], [16]. Other algorithms such as support vector machines, or SVM, [17], generalized instance sets [18], k-nearest neighbors algorithm [17], neural networks [19] and Generalized Discriminant Analysis, or GDA, [20] have been used to classify English text. Various algorithms have also been applied to other languages such as Chinese [17], [21] although there has been noticeably less research done with the Arabic language.

Most of those text classification algorithms handle text documents as a "bag-of-words" where a set of words or tokens are used to interpret the text and which rely on using their frequency in some manner [22], [23]. The traditional approach to text classification goes through four steps: first, pre-processing of the text where the words (or tokens) and sentences in the training files are segmented [1], [24]; second, using word/token counts to extract or select different features; thirdly, applying one of the machine learning algorithms mentioned earlier; and finally, performing the same feature extraction on the test data and applying the learned model to the extracted features to predict the class for the test data [1], [24].

During the process of analyzing the text, a complication occurs when the phenomenon of code-switching arises. This is where a text contains more than one language or variations of the same language. This phenomenon has been the subject of extensive linguistically oriented study in the past [25], [26]. This paper tackles the problem of mixed texts by segmenting those variations. Text segmentation is the task of automatically separating the text into identified or coherent parts [27]. Compared to text classification, text segmentation can be used to produce a more accurate estimate of each class, category or topic located inside the text rather than assigning a class or set of classes to the entire text as a whole.

يحضر برشلونة عرضا كبيرا لتعزيز الجبهة اليمنى التي تعاني بعد رحيل البرازيلي داني ألفيش إلى يوفنتوس ، وصرف في السنوات الأخيرة ١٦٧ مليون دولار لشراء مدافعين لم يكن أغلبهم على قدر التوقعات

Fig. 1. A sport news from aljazeera.net [66] in MSA text

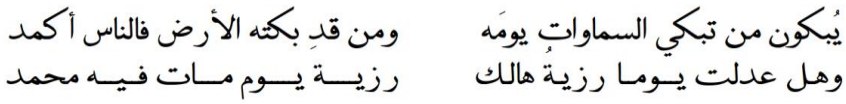


Fig. 2. A Classical Arabic Poem

Many segmentation algorithms, such as the TextTiling algorithm [28] and the dotplotting algorithm [29] rely on measuring the variation in word usage to predict potential boundaries in the text, where a vast difference in word usage is a positive sign. Kozima [30] introduced an algorithm that traces the coherence of a document by applying a semantic grid in a "lexical coherence profile". A statistical approach was proposed by [27] for text segmentation, where the algorithm builds a model from selected informative features, then the model is used to predict where boundaries happen in the text.

The work in this paper instead uses an approach based on the Prediction-by-Partial Matching (PPM) compression scheme as the basis of both text classification and segmentation. This Markov-based approach effectively uses character-based language models and has been employed in many NLP tasks in the past often with state-of-the-art results or results competitive with traditional schemes [1], [31]–[36].

Compared to the traditional way of text classification, compression-based language modelling is a character-based approach, whereas traditional text classification is a word-based approach which is language-dependent that tends to overlook both the contextual information of the text and the word order [1], [2]. The use of language modelling for text classification takes into consideration the contextual information in the text when building the language model and avoids the need for pre-processing of the text usually required by most classification algorithms [1], [2]. The use of Markov-based approximations standard in character-based language modelling avoids the issue of explicit feature selection that is applied in traditional classification and segmentation algorithms which may discriminate some important features of the text [1], [37]. The segmentation process performed in this study applies a Viterbi-style algorithm which produces an accurate estimate of each class, category or topic located in the text [34].

TABLE I. THE MOST UNIVERSALLY USED LANGUAGES

| Rank | Language | Users (millions) |
|---|---|---|
| 1 | Mandarin | 1051 |
| 2 | English | 508 |
| 3 | Hindi | 497 |
| 4 | Spanish | 392 |
| 5 | Arabic | 330 |
| 6 | Russian | 277 |
| 7 | Bengali | 211 |
| 8 | Portuguese | 191 |
| 9 | Malay | 159 |
| 10 | French | 129 |

The Arabic language "العربية" is acknowledged to be one of the most largely used languages, with 330 million people using the language as their first language, as shown in Table 1, plus 1.4 billion more using it as a secondary language [38]. The majority of the speakers are located across twenty-two nations, primarily in the Middle East, North Africa and Asia, and the United Nations considers the Arabic language as one of its five official languages. The Arabic language is part of the Semitic languages that includes Tigrinya, Amharic, Hebrew, etc., and shares almost the same structure as those languages. It has 28 letters, two genders – feminine and masculine, as well as singular, dual and plural forms. The Arabic language has a right-to-left writing system with the basic grammatical structure that consists of verb-subject-object and other structures, such as VOS, VO and SVO [39]–[41].

The non-colloquial written text for the Arabic language can be divided into two types: Classical Arabic and Modern Standard Arabic [42]–[45]. The Classical Arabic (CA) epoch is usually measured from the sixth century which is the start of Arabic literature. It is the language of the Holy Quran, the 1,400-year-old primary religious book of Islam with 77,430 words [46] and other ancient Islamic books from that era, such as the Hadith books [47]. With the beginning of journalism and the spread of literacy in the eighteenth century came Modern Standard Arabic or MSA. MSA is the language of the current printed Arabic media and most Arabic publications. (See Figure 1 for an example).

Almost all Arabic language NLP tasks are performed for MSA [48]. One example of those tasks is parts-of-speech tagging of Arabic language. Most of the popular Arabic parts-of-speech or POS taggers were trained on MSA text [36], [49], and therefore, the performance of the taggers will be best when tagging MSA text [36]. Contrastingly, tagging Classical Arabic text using MSA POS taggers will significantly reduce the quality of the tagging [47].

Some Arabic corpora, such as the Bangor Arabic Compression Corpus (BACC), is a mixture of both CA and MSA text. An example is the BACC sub-corpus *arabicbook1*, which contains both recent novels with ancient Arabic poems. (See Figure 2 for one example). The results of using such a corpus in order to perform various NLP tasks, such as POS tagging, as stated before, will vary and will not be consistent and reliable. Therefore, there arises a need to accurately classify CA from MSA within the text.

This paper explores the approach of classifying the Arabic text using PPM. It will first explain the PPM text compression scheme and its use for compressing, classifying and segmenting natural language text. Secondly, it will detail findings of PPM character-based modelling experiments used to classify and segment Arabic text. Thirdly, the results and

limitations of those experiments are discussed in detail. Finally, the conclusion is presented.

## II. Minimum Cross-Entropy as a Text Classifier

The PPM scheme uses an online algorithm that compresses the natural language adaptively as the text is processed sequentially. PPM usually processes character streams, although there are word-based and tag-based (i.e. using POS tags) variations. The standard character-based model uses the prior context of already received characters to foretell the imminent one. The word-based PPM model uses a series of already seen words to help predict the upcoming word but resorts to a character-based encoding for unseen words. The tag-based model, which is used in the last experiment, effectively encodes two streams, the stream of POS tags and the stream of words, by using the previously seen tags and words to predict the upcoming tag and word [50].

The basis of the classification and segmentation schemes in this paper use the character-based approach for compressing the Arabic text [1]. The essence of this approach depends on the concept of entropy as a measurement of the message's "information content" [51], and on the notion that the upper bound of the entropy can directly be estimated by compressing the text [52].

The fundamental coding theorem in information theory [51] states that an entropy of a sequence of text, or message, is the lower bound to the average number of bits per character required to encode that message [34].

$$H(P) = -\sum_{i=1}^{k} p\,(x_i) log\ p(x_i)$$

where there are *k* number of potential characters with the probability distribution $P = p(x_1),\ p(x_2),..., p(x_k)$ and the probabilities sum to 1 and are independent. The measurement of the uncertainty associated with the selection of the characters is represented by the entropy, where the higher the entropy, the higher the uncertainty. The message's "information content" can also be measured by the entropy, as the more probable the messages, the less information is conveyed compared to less probable ones [34].

A general case for a language with probability distribution can be extended from the previous equation for a text sequence $T = x_1, x_2, \ldots, x_m$ of length *m*:

TABLE II. Processing the String بجد/بجدبهوز/ using PPM

| Order 2 | | | Order 1 | | | Order 0 | | |
|---|---|---|---|---|---|---|---|---|
| Prediction | c | p | Prediction | c | p | Prediction | c | p |
| ‘جد’ → ب | 1 | 1/2 | ‘ج’ → د | 2 | 3/4 | → ج | 2 | 3/26 |
| → esc | 1 | 1/2 | → esc | 1 | 1/4 | → د | 2 | 3/26 |
| ‘دب’ → ه | 1 | 1/2 | ‘د’ → ب | 1 | 1/2 | → و | 1 | 1/26 |
| → esc | 1 | 1/2 | → esc | 1 | 1/2 | → ب | 4 | 7/26 |
| ‘وب’ → أ | 1 | 1/2 | ‘و’ → ب | 1 | 1/2 | → ه | 1 | 1/26 |
| → esc | 1 | 1/2 | → esc | 1 | 1/2 | → أ | 3 | 5/26 |
| ‘بج’ → د | 2 | 3/4 | ‘ب’ → ج | 2 | 3/8 | → esc | 6 | 3/13 |
| → esc | 1 | 1/4 | → ه | 1 | 1/8 | | | |
| ‘به’ → و | 1 | 1/2 | → أ | 1 | 1/8 | Order -1 | | |
| → esc | 1 | 1/2 | → esc | 3 | 3/8 | Prediction | c | p |
| ‘با’ → أ | 1 | 1/2 | ‘ه’ → و | 1 | 1/2 | → A | 1 | 1/\|A\| |
| → esc | 1 | 1/2 | → esc | 1 | 1/2 | | | |
| ‘هو’ → ب | 1 | 1/2 | ‘أ’ → ب | 2 | 1/2 | | | |
| → esc | 1 | 1/2 | → أ | 1 | 1/6 | | | |
| ‘أب’ → ج | 2 | 3/4 | → esc | 2 | 1/3 | | | |
| → esc | 1 | 1/4 | | | | | | |
| ‘أأ’ → ب | 1 | 1/2 | | | | | | |
| → esc | 1 | 1/2 | | | | | | |

$$H(L) = \lim_{m\to\infty} -\frac{1}{m}\sum p\,(x_1, x_2, \ldots, x_m)\ log\ p(x_1, x_2, \ldots, x_m).$$

This describes the entropy of a language which is defined to be the limit of the entropy when the size of the message becomes large. The probability distribution for the source language *L* is usually not identified or known. Nevertheless, applying a model *M* as an approximation to the probability distribution gives the upper bound to $H(L)$ [34]:

$$H(L, M) = -\sum P_M\,(x_1, x_2, \ldots, x_m)\ log\ P_M(x_1, x_2, \ldots, x_m)$$

where $P_M(x_1, x_2, \ldots, x_m)$ is used to estimate the probabilities. $H(L, M)$ is described as the *cross-entropy* which is higher than or equivalent to the entropy $H(L)$, as this is based on the source itself which is the best possible language model:

$$H(L) \leq H(L, M).$$

Compressing the text can be used to estimate an upper bound to the entropy of a message [52]. Considering the number of bits needed to encode a sequence of text to be $b_M(x_1, x_2, \ldots, x_m)$, when using some model $M$ to estimate the probabilities, then:

$$H(L, M, T) = \lim_{m \to \infty} \frac{1}{m} b_M(x_1, x_2, \ldots, x_m)$$

where the number of bits per character needed to encode a long text message $T$ formed from $L$ is $H(L, M, T)$.

The cross-entropy is important as it presents a measurement of how great the estimated model is performing on the test text; the less inexact the model is, the closer the cross-entropy is to $H(L)$. Furthermore, by measuring the cross-entropy for every possible model, the cross-entropy provides a valuable measure for analysing the correctness of the competing models. The model that has the least cross-entropy is judged to be the "best" or most appropriate. The information is derived from a semantic label which is associated with each model which reflects the class or type of data that was used to train the model. Simply, the label linked with the "best" model is selected and used to classify the text:

$$\hat{\theta}(T) = \text{argmin}_i \, H(L, M, T).$$

The following section presents one specific method, which is based on the PPM text compression, used to measure the cross-entropy [34].

## III. PPM-based Compression for Natural Language Text

PPM is an adaptive online system used for compressing text by predicting the upcoming symbol or character using the prior context with a given maximum fixed length. It applies a Markov-based *n*-gram method which utilises a back-off mechanism similar to that proposed by Katz [53]. Nonetheless, backing-off is referred by PPM as "escaping" and it was developed before the mechanism that Katz proposed. Cleary and Witten [54] first proposed PPM in 1984 when they described the two character-based PPM variants, PPMA and PPMB. Two further character-based variants of PPM, PPMC and PPMD, were introduced by Moffat and Howard in 1990 and 1993 respectively [55]. The main difference among these four versions of PPM is the calculation of the escape probability which the smoothing mechanism needs for backing off to a model's lower order. Many trials have been performed on character streams which have shown that PPMD ordinarily gives better compression results when compared with the results of other PPM variants [56].

As mentioned before, PPM has been successfully implemented in various fields of NLP. It produces state-of-the-art text compression results for many languages as detailed in the reports mentioned in [31], [36], [57]. PPM has been used as the basis for an effective method for performing Chinese word segmentation where spaces (as word separators) are inserted into Chinese text which has no spaces [33]. Other studies such as [31], [34]–[36], [57], [58] have reported using PPM for different languages for other NLP tasks such as cryptology, code switching, authorship attribution, text correction and speech recognition.

The next equation defines the probability $p$ for the next character $\varphi$ when using PPMD [34]:

$$p(\varphi) = \frac{2c_d(\varphi) - 1}{2T_d} \quad (1)$$

where $d$ represent the currently used coding order, the total amount of times that the current context has occurred is indicated by $T_d$ and $c_d(\varphi)$ represents the total number of occurrences for the symbol $\varphi$ in the current context. The estimation of the escape probability $e$ by PPMD is as follows:

$$e = \frac{t_d}{2T_d} \quad (2)$$

where the total amount of times that a unique character has occurred following the current context is represented by $t_d$. Most experiments have reported that using the maximum order of 5 to be the most efficient, as PPMD starts with that order first to encode the incoming character [34] before escaping down to lower orders if necessary.

If the forthcoming symbol was predicted by the current model and the model contained it, then its probability in the current maximum order, 5 in this case, will be used to transmit it. If the forthcoming symbol was not found in the model, then the encoder will escape to the next lower order model, 4 in this case. This process of escaping will be repeated until the model finds that symbol or prediction. If the model does not contain the symbol, then the encoder will back off to a default order of -1 where the same probability is used for all symbols in the alphabet [34].

To describe character-based encoding in further detail, the way PPM models a given sequence of text is presented in Table 2. This example uses a specific variant of PPM prediction method, PPMD, to model the string أبجدبهوياً/أبجد. As stated, a model's maximum order of 5 is proven to be efficient, but a maximum model order of 2 is used in this example for illustration purposes. In the table, *c* shows the count, *p* expresses the probability and the size of the alphabet used is represented by $|A|$ [33]. For this example, let the next character be letter ب. This character has been seen once before ('جد' → ب) for the order two context 'جد' and consequently it has a probability of ½ (utilising equation (1) as the count is 1). Accordingly, the encoder will use 1 bit to encode the character ب. However, if the forthcoming character in the order two context had not been seen before, (i.e. assume the following letter was ا rather than ب), an escape process to a lower order will be performed, the escape probability will be ½ (from equation (2)), and the model will use a lower order of 1.

During the process of the model's backing off, the new order will be adopted to calculate the probability, and in this instance, there is no character ا that appears after د. Consequently, another escape probability of $\frac{1}{2}$ will be encoded by the model, and the current context is decreased to the order 0 context (the null context). This order, where the probability will be $\frac{5}{26}$, is used to encode letter ا. The whole cost of foretelling the last letter is $\frac{5}{26} \times \frac{1}{2} \times \frac{1}{2}$, which costs the encoder

approximately 4.4 bits to encode it. Furthermore, if the next character has not been seen already in the context, for example letter $\xi$, starting from the maximum order of 2 and escaping down to -1, the model encodes the escape probability three times using the following probabilities: $\frac{1}{2} \times \frac{1}{2} \times \frac{3}{13} \times \frac{1}{256}$. This is because order -1 is applied to encode this character and using an 8-bit encoding for the Arabic text (say), the alphabet size is 256. This results in approximately 12.1 bits being required to encode it [34].

(PPM when used for compression uses arithmetic coding [59] to ensure that the cost of encoding is close to the theoretical cost of $\log_2 p$ bits for some probability *p*, although physical coding is not required for the classification and segmentation applications used here since the theoretical encoding cost can be computed directly instead without the need for physical encoding tor these applications).

## IV. PPM Classifier and Segmenter Evaluation Experiments

This section reports the experiments that were performed as part of the evaluation of the PPM classifier and segmenter when applied to Arabic text. Four experiments were conducted: (A) initial classification experiments; (B) classification of mixed Arabic corpora; (C) segmentation of mixed Arabic corpora; and (D) tagged-based compression experiments of Arabic text. The first experiment used 200 files for the initial evaluation process. The second experiment examined the result of classifying a number of Arabic mixed corpora. The third experiment performed text segmentation on the same Arabic mixed corpora using a Viterbi-style algorithm that finds the most probable sequence of segmented characters. Lastly, the final experiment conducted tag-based compression experiments using the previous outcomes. The Arabic mixed corpora which were used were the Bangor Arabic Compression Corpus (BACC) [57], the Universal Dependencies (UD) project corpus [60], Arabic in Business and Management corpus (ABMC) [61] and the Arabic Learner Corpus [62].

Segmenting Arabic text increases the performance of some NLP applications such as parts-of-speech tagging. As stated before, most Arabic NLP tasks are trained and built for MSA. The performance of such a task drops when applied to Classical text [47]. The object of the research described here is to classify and segment CA and MSA using the PPM character-based compression algorithm. The experiments in this paper used two language models, one for CA and another for MSA. Published Arabic corpora that contain mostly the required type of Arabic text were used to train the two static models.

The MSA model was trained using Corpus A [58]. The corpus was recently published, and most of its genres covers several current MSA areas such as business, cinema, opinions, conferences, economics, politics and more. The text in the corpus was collected from the bilingual newspaper Al-Hayat website, and from the open-source online corpus, OPUS [63].

The second model used in this paper was trained using CA text from the King Saud University Corpus of Classical Arabic (KSUCCA). According to the author [64], the foremost purpose of the creation of this corpus is for analysing the lexical meaning of the Holy Quran. The corpus is relatively large and it contains over 50 million words, split into six genres such as Literature, Linguistics and Science. To generate a relatively similar size as the first model (as this helps improve classification accuracy), the sub-genre Religion was not included in the training process. To obtain a more robust evaluation and ensure the training text used for the models was separate from the testing text, a tenfold cross-validation technique was used for the classification experiments.

Both the PPM language modelling and the segmentation were performed using the Text Mining Toolkit described in [65]. This toolkit allows *static* models to be created from training text. That is, once the models have been created, they can be used to prime the model(s) used by the application and are subsequently not altered during the compression, classification or segmentation processes.

### A. Initial Classification Experiments

This initial experiment was conducted to evaluate the PPM classifier based on four evaluation criteria and using 200 test files in the evaluation process. The testing files were divided into two groups, each with 100 files. The first group comprised 100 files that contained CA text randomly gathered from the Holy Quran, Islamic books such as Ibn Qayyim and Ahmad ibn Hanbal and poems from the famous Arab poet, Al-Mutanabbi. The second group comprised 100 files containing MSA text randomly collected from popular Arabic news websites such Aljazeera.net [66], BBC Arabic [67] and skynewsarabia [68] and recently published novels.

Four evaluation criteria (Accuracy, Recall, Precision and F-measure) were used to evaluate the classification results using the following equations:

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

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where *TP* is the true positives which are the number of cases where the prediction matches the type of Arabic text and *TN* is the true negatives which represent the number cases where the prediction does not match the type of Arabic text, and *FP* and *FN* are the false positives and false negatives respectively, as shown for the confusion matrix in Table 3.

TABLE III. Confusion Matrix

| | Predicted CA | Predicted MSA |
|---|---|---|
| Actual CA | TN | FP |
| Actual MSA | FN | TP |

Classifying the Classical and MSA text using the PPM compression algorithm obtained an accuracy of 95.5%, an

average precision of 0.958, an average recall of 0.955 and an average F-measure of 0.954. The results in Table 4 show that the PPM classifier predicted all the 100 files that contain CA text and 91 out of 100 files which have MSA text.

TABLE IV. PPM CLASSIFICATION RESULTS

| | Predicted CA | Predicted MSA |
|---|---|---|
| Actual CA | 100 | 0 |
| Actual MSA | 9 | 91 |

TABLE V. CLASSIFICATION RESULTS OF UD

| Genre | Corpus Size | Classical model Compression (bytes) | Modern model Compression (bytes) | Classical bpc | Modern bpc | Predicted Type |
|---|---|---|---|---|---|---|
| **AFP** | 138,223 | 35,788 | 33,149 | 2.07 | **1.92** | MSA |
| **UMH** | 426,811 | 106,478 | 97,517 | 2.00 | **1.83** | MSA |
| **XIN** | 158,997 | 40,660 | 36,709 | 2.05 | **1.85** | MSA |
| **ALH** | 108,599 | 27,419 | 25,536 | 2.02 | **1.88** | MSA |
| **ANN** | 130,068 | 32,847 | 31,227 | 2.02 | **1.92** | MSA |
| **XIA** | 293,104 | 74,650 | 67,550 | 2.04 | **1.84** | MSA |

TABLE VI. CLASSIFICATION RESULTS OF ABMC

| Genre | Corpus Size | Classical model Compression (bytes) | Modern model Compression (bytes) | Classical bpc | Modern bpc | Predicted Type |
|---|---|---|---|---|---|---|
| **Economic News** | 2,201,462 | 544,181 | 496,183 | 1.98 | **1.80** | MSA |
| **Management** | 1,358,576 | 317,477 | 275,826 | 1.87 | **1.62** | MSA |
| **Stock News** | 890,493 | 224,493 | 199,571 | 2.02 | **1.79** | MSA |

### *B. Classifying Mixed Arabic Corpora*

Classifying corpora of unknown origins, or for which it may be suspected may have a mixture of CA and MSA text, will help Arabic NLP researchers to confirm their content. The experiment reported in this section investigated whether it was possible to accurately perform a document level text classification of some Arabic corpora. Table 5 displays the results of this experiment for the UD corpus, Table 6 for the ABMC, Table 7 for the Arabic Learner Corpus and Table 8 for the BACC corpus. The tables list the size of the files compressed files, the size of the compressed output files (in bytes), the compression ratios (in bits per character or 'bpc') and the type (CA or MSA) predicted from the model with the best compression.

TABLE VII. CLASSIFICATION RESULTS OF ARABIC LEARNER CORPUS

| Genre | Corpus Size | Classical model Compression (bytes) | Modern model Compression (bytes) | Classical bpc | Modern bpc | Predicted Type |
|---|---|---|---|---|---|---|
| **Arabic Learner Corpus** | 2,806,467 | 620,563 | 630,306 | **1.77** | 1.80 | CA |

TABLE VIII. CLASSIFICATION RESULTS OF BACC

| Genre | Corpus Size | Classical model Compression (bytes) | Modern model Compression (bytes) | Classical bpc | Modern bpc | Predicted Type |
|---|---|---|---|---|---|---|
| **arabicbook1** | 829,036 | 187,362 | 192,804 | **1.81** | 1.86 | CA |
| **arabicbook2** | 884,273 | 202,343 | 206,271 | **1.83** | 1.87 | CA |
| **arabicbook3** | 977,286 | 223,451 | 229,887 | **1.83** | 1.88 | CA |
| **arabichistory** | 30,251,137 | 5,750,445 | 7,838,286 | **1.52** | 2.07 | CA |
| **arabiclitriture** | 18,594,383 | 3,846,029 | 4,877,075 | **1.65** | 2.10 | CA |
| **arabicpoems** | 46,929 | 11,701 | 12,313 | **1.99** | 2.10 | CA |
| **artandmusic** | 41,770 | 9,665 | 9,137 | 1.85 | **1.75** | MSA |
| **articles** | 101,641 | 22,982 | 21,630 | 1.81 | **1.70** | MSA |
| **bookcollection** | 197,935,882 | 40,631,602 | 48,551,255 | **1.64** | 1.96 | CA |
| **culture** | 34,188 | 7,867 | 7,363 | 1.84 | **1.72** | MSA |
| **economic** | 15,352 | 3,583 | 3,066 | 1.87 | **1.60** | MSA |
| **education** | 26,418 | 6,078 | 5,504 | 1.84 | **1.67** | MSA |
| **political** | 46,884 | 10,995 | 9,785 | 1.88 | **1.67** | MSA |
| **press** | 536,692 | 122,961 | 111,260 | 1.83 | **1.66** | MSA |
| **sports** | 31,059 | 7,225 | 6,659 | 1.86 | **1.72** | MSA |
| **stories** | 1,022,476 | 242,699 | 237,372 | 1.90 | **1.86** | MSA |

The steps of the experiment are as follows:

- Using the two static models created earlier for priming, two compressed files are generated by compressing the Arabic texts using an order 5 PPMD character-based compression scheme.
- Then, the cross-entropy or the size of the two compressed files are compared and the class label of the

text, in this case CA or MSA, is chosen from the file with the smallest compressed size.

The classification results from these tables show that the sub-genre of some Arabic corpora, such as the BACC in Table 8, contain different types of Arabic text. Other corpora, such as the Arabic Learner Corpus in Table 7, have similar compression sizes which provides an indication that they contains mixed text of both CA and MSA.

TABLE IX. SEGMENTATION RESULTS OF UD

| Data set | Number of words | Number of Classical words | Number of Modern words | Classical (CA) % | Modern (MSA) % |
|---|---|---|---|---|---|
| **AFP** | 11,369 | 594 | 10,775 | 5.22% | 94.78% |
| **UMH** | 34,765 | 2,053 | 32,712 | 5.91% | 94.09% |
| **XIN** | 12,666 | 554 | 12,112 | 4.37% | 95.63% |
| **ALH** | 9,019 | 1,078 | 7,941 | 11.95% | 88.05% |
| **ANN** | 11,152 | 2,252 | 8,900 | 20.19% | 79.81% |
| **XIA** | 23,930 | 617 | 23,313 | 2.58% | 97.42% |

<Classic> به إلا ضربا من الحلم أو الكابوس حدقت به فاضت عيناها واصفرت وجنتاها ارتجفت أصابعها<\Classic>
<Modern> أنت تريد بناء دولة خارج الزمان والمكان وتترك فلسطين لماذا لا تقدم هذا الاقتراح إلى إدارة السجن<\Modern>
<Classic> الحسنة وادفع الشبهات الغليظة النتنة والأفكار الباطلة وأخرج الناس من ظلمات الجهلة إلى نور النيرة <\Classic>
<Modern> كنت مع زملائي سفرت إلى الرياض للدراسة عندما وصلت إلى المطار رأيت أجانب من البلاد المختلفة <\Modern>

Fig. 3. Random segmented samples from The BACC

TABLE X. SEGMENTATION RESULTS OF ABMC

| Genre | Number of words | Number of Classical words | Number of Modern words | Classical (CA) % | Modern (MSA) % |
|---|---|---|---|---|---|
| **Economic News** | 169,374 | 12,200 | 157,174 | 7.20% | 92.80% |
| **Management** | 121,603 | 7,192 | 114,411 | 5.91% | 94.09% |
| **Stock News** | 87,943 | 53 | 87,890 | 0.06% | 99.94% |

### *C. Segmenting Mixed Arabic Corpora*

The compression results from the last classification experiment show that some Arabic corpora contain mixed CA and MSA text. To produce a more accurate estimate of CA and MSA text, this experiment performed a text segmentation using a Viterbi-style algorithm that finds the most probable sequence of characters of each class, category or topic in the text [33] where all possible switching of encoding models are considered.

This experiment was conducted as follows:

- The same Text Mining Toolkit was used to segment the text file at the character level to insert labels (tags), either CA or MSA, inside the text. The segmentation in this step was applied using the Viterbi-style algorithm [33]. Figure 3 shows a sample from the segmented files.
- Then, a post-processing of the resulting file was performed to count all the terms of each label and for separating the two types of text into two files for additional experiments.

Table 9 displays the outcomes of this experiment for the UD corpus, Table 10 for the ABMC, Table 11 for the Arabic Learner Corpus and Table 12 for the BACC corpus. The tables list the numbers of words in the segmented files for both Classical (CA) and Modern (MSA) texts and the percentages of each.

The results from the previous tables indicate that some Arabic corpora contain mixed CA and MSA text, and the PPM compression models can be used to produce an accurate estimate of the extent of both Arabic text types. The illustration of the segmentation process is shown in Figure 3 which shows randomly selected segmented samples from two of BACC sub-genre, *arabiclearnercorpus* and *arabicbook1*. The sample demonstrates typical output of the segmentation process which produces a more accurate picture of the textual contents.

TABLE XI. SEGMENTATION RESULTS OF ARABIC LEARNER CORPUS

| Genre | Number of words | Number of Classical words | Number of Modern words | Classical (CA) % | Modern (MSA) % |
|---|---|---|---|---|---|
| **Arabic Learner Corpus** | 287,107 | 161,897 | 125,210 | 56.39% | 43.61% |

### *D. Tag-based Compression Experiments*

Most Arabic language NLP tasks are made for processing MSA [48], and POS tagging of Arabic language is one example of those tasks. Since most popular recognised Arabic POS taggers were trained on MSA text [36], [49], the tagging of mixed corpora text will vary in quality and will not be consistent and reliable. The final experiment was conducted using tag-based compression on mixed Arabic corpora, selected using the previous outcomes in order to evaluate the process of both the tagging and classification.

The experiment was performed as follows:

- First, both CA and MSA files identified from experiments (B) and (C) were tagged using the MADAMIRA [69] Arabic tagger.
- The tagged files were then post-processed where terms and tags were extracted into one file for the next step.
- The tagged files were then compressed using a tag-based compression scheme and the results compared with the compressed files created using a character-

based compression scheme to evaluate the compression quality.

TABLE XII. SEGMENTATION RESULTS OF BACC

| Genre | Number of words | Number of Classical words | Number of Modern words | Classical (CA) % | Modern (MSA) % |
|---|---|---|---|---|---|
| **arabicbook1** | 85,441 | 65,867 | 19,574 | 77.09% | 22.91% |
| **arabicbook2** | 89,015 | 61,645 | 27,370 | 69.25% | 30.75% |
| **arabicbook3** | 104,055 | 83,503 | 20,552 | 80.25% | 19.75% |
| **arabichistory** | 3,350,365 | 3,348,513 | 1,852 | 99.94% | 0.06% |
| **arabiclitriture** | 1,983,790 | 1,978,670 | 5,120 | 99.74% | 0.26% |
| **arabicpoems** | 4,701 | 4,151 | 550 | 88.30% | 11.70% |
| **artandmusic** | 3,985 | 528 | 3,457 | 13.25% | 86.75% |
| **articles** | 9,624 | 1,792 | 7,832 | 18.62% | 81.38% |
| **bookcollection** | 20,725,720 | 19,836,491 | 889,229 | 95.71% | 4.29% |
| **culture** | 3,107 | 476 | 2,631 | 15.32% | 84.68% |
| **economic** | 1,376 | 3 | 1,373 | 0.22% | 99.78% |
| **education** | 2,437 | 33 | 2,404 | 1.35% | 98.65% |
| **political** | 4,317 | 62 | 4,255 | 1.44% | 98.56% |
| **press** | 50,977 | 4,351 | 46,626 | 8.54% | 91.46% |
| **sports** | 2,875 | 221 | 2,654 | 7.69% | 92.31% |
| **stories** | 111,809 | 28,664 | 83,145 | 25.64% | 74.36% |

Table 13 lists the results of the experiment which shows in the second column the percentage improvement in compression for the tab-based compression scheme over the character-based compression scheme, and the type of text (CA or MSA) in the third column that was confirmed in the earlier experiments. A positive percentage improvement indicates the tag-based compression was better, and a negative improvement indicates the character-based compression was better.

The results in Table 13 show that utilising the tags to compress the BACC sub-corpus *'arabicliteriture'*, which was found to consist of 99.74% Classical Arabic text, decreases the compression percentage by 4.38% (compared with the character based compression scheme). However, using the same compression model to compress the ABMC sub-corpus 'Economic-News', which was found to consist of 92.80% MSA text, increases the compression percentage by 6.50% (compared with the character based compression scheme). The difference in compression quality provides an indication that the quality of tagging for the CA text has dropped, compared to the quality of tagging for the MSA text, because the compression size has increased.

TABLE XIII. TAG-BASED COMPRESSION ON CA AND MSA TEXT

| Corpus | Tag-based Compression Improvement | Text Type |
|---|---|---|
| **BACC - arabichistory** | -5.07% | CA |
| **BACC – arabiclitriture** | -4.38% | CA |
| **BACC – bookcollection** | -3.56% | CA |
| **ABMC – Economic News** | 6.50% | MSA |
| **Corpus A – books** | 2.76% | MSA |

## V. CONCLUSION

Classification of Classical Arabic (CA) and Modern Standard Arabic (MSA) text was performed on sample texts using a PPM character-based compression scheme achieving an accuracy of 95.5%, an average precision of 0.958, an average recall of 0.955 and an average F-measure of 0.954. Three further experiments were implemented in this study to analyse mixed Arabic corpora. First, a classification of Arabic corpora was performed and the results showed that different sub-genres of some Arabic corpora contain different types of Arabic text since the compression size for other corpora indicated that the texts were a mixture between CA and MSA. Then a segmentation of the same corpora was accomplished using a Viterbi-based algorithm and the results indicated that segmenting the text produces a more accurate estimate of CA and MSA text. Finally, tag-based compression experiments (using parts-of-speech taggers) were performed to evaluate the tagging quality and the results showed a difference in compression quality between CA and MSA texts. This provides an indication that the quality of the tagging is affected when either CA and MSA text is being tagged therefore showing that NLP applications (such as taggers) should treat these texts separately and use different training data for each or process them differently.

## REFERENCES

[1] W. J. Teahan and D. J. Harper, "Using compression-based language models for text categorization," in Language modeling for information retrieval, Springer, 2003, pp. 141–165.

[2] H. Ta'amneh, E. A. Keshek, M. B. Issa, M. Al-Ayyoub, and Y. Jararweh, "Compression-based arabic text classification," in Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, 2014, pp. 594–600.

[3] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in Mining text data, Springer, 2012, pp. 163–222.

[4] P. Juola and others, "Authorship attribution," Found. Trends{®} Inf. Retr., vol. 1, no. 3, pp. 233–334, 2008.

[5] E. Stamatatos, "A survey of modern authorship attribution methods," J. Am. Soc. Inf. Sci. Technol., vol. 60, no. 3, pp. 538–556, 2009.

[6] A. Alwajeeh, M. Al-Ayyoub, and I. Hmeidi, "On authorship authentication of arabic articles," in Information and Communication Systems (ICICS), 2014 5th International Conference on, 2014, pp. 1–6.

[7] N. Cheng, R. Chandramouli, and K. P. Subbalakshmi, "Author gender identification from text," Digit. Investig., vol. 8, no. 1, pp. 78–88, 2011.

[8] K. Alsmearat, M. Al-Ayyoub, and R. Al-Shalabi, "An extensive study of the bag-of-words approach for gender identification of arabic articles," in Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, 2014, pp. 601–608.

[9] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," ACM Trans. Inf. Syst., vol. 26, no. 3, p. 12, 2008.

[10] N. Abdulla, N. Mahyoub, M. Shehab, and M. Al-Ayyoub, "Arabic sentiment analysis: Corpus-based and lexicon-based," in Proceedings of The IEEE conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2013.

[11] M. N. Al-Kabi, N. A. Abdulla, and M. Al-Ayyoub, "An analytical study of arabic sentiments: Maktoob case study," in Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for, 2013, pp. 89–94.

[12] O. F. Zaidan and C. Callison-Burch, "The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, 2011, pp. 37–41.

[13] S. Malmasi, E. Refaee, and M. Dras, "Arabic dialect identification using a parallel multidialectal corpus," in International Conference of the Pacific Association for Computational Linguistics, 2015, pp. 35–53.

[14] V. Korde and C. N. Mahender, "Text classification and classifiers: A survey," Int. J. Artif. Intell. Appl., vol. 3, no. 2, p. 85, 2012.

[15] S. Eyheramendy, D. D. Lewis, and D. Madigan, "On the naive bayes model for text categorization," 2003.

[16] F. Peng, D. Schuurmans, and S. Wang, "Augmenting naive bayes classifiers with statistical language models," Inf. Retr. Boston., vol. 7, no. 3–4, pp. 317–345, 2004.

[17] J. He, A.-H. Tan, and C.-L. Tan, "On machine learning methods for Chinese document categorization," Appl. Intell., vol. 18, no. 3, pp. 311–322, 2003.

[18] W. Lam and Y. Han, "Automatic textual document categorization based on generalized instance sets and a metamodel," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 5, pp. 628–633, 2003.

[19] M. E. Ruiz and P. Srinivasan, "Hierarchical neural networks for text categorization (poster abstract)," in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 281–282.

[20] T. Li, S. Zhu, and M. Ogihara, "Efficient multi-way text categorization via generalized discriminant analysis," in Proceedings of the twelfth international conference on Information and knowledge management, 2003, pp. 317–324.

[21] J.-J. Tsay and J.-D. Wang, "Improving linear classifier for Chinese text categorization," Inf. Process. Manag., vol. 40, no. 2, pp. 223–237, 2004.

[22] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," Appl. Intell., vol. 19, no. 1–2, pp. 109–123, 2003.

[23] F. Sebastiani, "Machine learning in automated text categorization," ACM Comput. Surv., vol. 34, no. 1, pp. 1–47, 2002.

[24] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in Proceedings of the seventh international conference on Information and knowledge management, 1998, pp. 148–155.

[25] P. Gardner-Chloros, Code-switching. Cambridge University Press, 2009.

[26] H. Schendl and L. Wright, Code-switching in early English, vol. 76. Walter de Gruyter, 2012.

[27] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," Mach. Learn., vol. 34, no. 1–3, pp. 177–210, 1999.

[28] M. A. Hearst, "Multi-paragraph segmentation of expository text," in Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994, pp. 9–16.

[29] J. C. Reynar, "An automatic method of finding topic boundaries," in Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994, pp. 331–333.

[30] H. Kozima, "Text segmentation based on similarity between words," in Proceedings of the 31st annual meeting on Association for Computational Linguistics, 1993, pp. 286–288.

[31] W. J. Teahan, "Modelling English Text," Ph.D. thesis, Waikato University, 1998.

[32] W. J. Teahan and J. G. Cleary, "Applying compression to natural language processing," in SPAE: The Corpus of Spoken Professional American-English., 1997.

[33] W. J. Teahan, Y. Wen, R. McNab, and I. H. Witten, "A compression-based algorithm for Chinese word segmentation," Comput. Linguist., vol. 26, no. 3, pp. 375–393, 2000.

[34] W. J. Teahan, "Text classification and segmentation using minimum cross-entropy," in Content-Based Multimedia Information Access-Volume 2, 2000, pp. 943–961.

[35] N. R. Al-Kazaz, S. A. Irvine, and W. J. Teahan, "An Automatic Cryptanalysis of Transposition Ciphers Using Compression," in International Conference on Cryptology and Network Security, 2016, pp. 36–52.

[36] I. S. Alkhazi, M. A. Alghamdi, and W. J. Teahan, (in press) "Tag based models for Arabic Text Compression."

[37] T. Joachims, Learning to classify text using support vector machines: Methods, theory and algorithms. Kluwer Academic Publishers, 2002.

[38] A. Soudi, A. Farghaly, G. Neumann, and R. Zbib, Challenges for Arabic machine translation, vol. 9. John Benjamins Publishing, 2012.

[39] M. A. Alghamdi, I. S. Alkhazi, and W. J. Teahan, "Arabic OCR Evaluation Tool," in Computer Science and Information Technology (CSIT), 2016 7th International Conference on, 2016, pp. 1–6.

[40] S. Green and C. Manning, "Better Arabic parsing: Baselines, evaluations, and analysis," COLING '10 Proc. 23rd Int. Conf. Comput. Linguist., no. August, pp. 394–402, 2010.

[41] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," 2008.

[42] P. Damien, N. Wakim, and M. Egea, "Phoneme-viseme mapping for Modern, Classical Arabic language," in Advances in Computational Tools for Engineering Applications, 2009. ACTEA'09. International Conference on, 2009, pp. 547–552.

[43] M. M. Najeeb, A. A. Abdelkader, and M. B. Al-Zghoul, "Arabic natural language processing laboratory serving Islamic sciences," Int. J. Adv. Comput. Sci. Appl., vol. 5, no. 3, 2014.

[44] K. C. Ryding, A reference grammar of modern standard Arabic. Cambridge university press, 2005.

[45] M. A. Alghamdi and W. J. Teahan, "A New Thinning Algorithm for Arabic Script," Int. J. Comput. Sci. Inf. Secur., vol. 15, no. 1, p. 204, 2017.

[46] K. Dukes and N. Habash, "Morphological Annotation of Quranic Arabic.," in LREC, 2010.

[47] M. S. Alrabiah, "Building A Distributional Semantic Model for Traditional Arabic and Investigating its Novel Applications to The Holy Quran," Ph.D. thesis, King Saud University, 2014.

[48] K. Dukes, "Statistical parsing by machine learning from a Classical Arabic treebank," Ph.D. thesis, University of Leeds, 2013.

[49] M. Maamouri, A. Bies, T. Buckwalter, H. Jin, and W. Mekki, "Arabic Treebank: Part 3 (full corpus) v 2.0 (MPG + Syntactic Analysis)," LDC2005T20, 2005. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2005T20. [Accessed: 25-Nov-2016].

[50] W. J. Teahan and J. G. Cleary, "Tag Based Models of English Text.," in Data Compression Conference, 1998, pp. 43–52.

[51] C. E. Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. J., vol. 27, pp. 623--656, 1948.

[52] P. F. Brown, V. J. Della Pietra, R. L. Mercer, S. A. Della Pietra, and J. C. Lai, "An estimate of an upper bound for the entropy of English," Comput. Linguist., vol. 18, no. 1, pp. 31–40, 1992.

[53] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," IEEE Trans. Acoust., vol. 35, no. 3, pp. 400–401, 1987.

[54] I. Cleary, John and Witten, "Data compression using adaptive coding and partial string matching," vol. C, no. 4, pp. 396–402, 1984.

[55] P. Wu, "Adaptive models of Chinese text," Ph.D. thesis, Bangor University, 2007.

[56] D. V Khmelev and W. J. Teahan, "A repetition based measure for verification of text collections and for text categorization," in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003, pp. 104–110.

[57] K. M. Alhawiti, "Adaptive models of Arabic text," Ph.D. thesis, Bangor University, 2014.

[58] S. Alkahtani and W. J. Teahan, "Aligning a New Parallel Corpus of Arabic-English," in Proceedings of the Eighth Saudi Students Conference in the UK, 2015, p. 279.

[59] J. G. Cleary, W. J. Teahan, and I. H. Witten, "Unbounded length contexts for PPM," in Data Compression Conference, 1995. DCC'95. Proceedings, 1995, pp. 52–61.

[60] M. J. A. ˇZeljko Agic et al., "Universal dependencies 1.1," LINDAT/CLARIN Digit. Libr. Inst. Form. Appl. Linguist. Charles Univ. Prague, 2015.

[61] M. EL-Haj, "Arabic in Business and Management Corpora (ABMC)." [Online]. Available: http://www.lancaster.ac.uk/staff/elhaj/corpora.htm. [Accessed: 27-Mar-2017].

[62] A. Y. G. Alfaifi, "Building the Arabic Learner Corpus and a System for Arabic Error Annotation," Ph.D. thesis, University of Leeds, 2015.

[63] S. Alkahtani, "Building and verifying parallel corpora between Arabic and English," Ph.D. thesis, Bangor University, 2015.

[64] M. Alrabiah, A. Al-Salman, and E. S. Atwell, "The design and construction of the 50 million words KSUCCA," in Proceedings of WACL'2 Second Workshop on Arabic Corpus Linguistics, 2013, pp. 5–8.

[65] T. C. Bell, J. G. Cleary, and I. H. Witten, Text Compression. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

[66] "الجزيرة.نت." [Online]. Available: http://www.aljazeera.net/portal. [Accessed: 18-Mar-2017].

[67] "BBC Arabic - الأوسط الشرق أخبار." [Online]. Available: http://www.bbc.com/arabic/middleeast. [Accessed: 18-Mar-2017].

[68] "عربية نيوز سكاي | اليوم أخبار." [Online]. Available: http://www.skynewsarabia.com/web/home. [Accessed: 23-Mar-2017].

[69] A. Pasha et al., "MADAMIRA : A Fast , Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," Proc. 9th Lang. Resour. Eval. Conf., pp. 1094–1101, 2014.