View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by White Rose Research Online

UNIVERSITY OF LEEDS

This is a repository copy of *Compression versus traditional machine learning classifiers to detect code-switching in varieties and dialects: Arabic as a case study*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/155881/

Version: Accepted Version

Article:

Tarmom, T orcid.org/0000-0002-2834-461X, Teahan, W, Atwell, E orcid.org/0000-0001-9395-3764 et al. (1 more author) (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. ISSN 1351-3249

https://doi.org/10.1017/S135132492000011X

© Cambridge University Press 2020. This article has been published in a revised form in Natural Language Engineering [https://doi.org/10.1017/S135132492000011X]. This version is free to view and download for private research and study only. Not for re-distribution, re-sale or use in derivative works. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose
university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

© 1998 Cambridge University Press

Compression vs Traditional Machine Learning Classifiers to Detect Code-switching in Varieties and Dialects: Arabic as a Case Study

TAGHREED TARMOM
School of Computing, University of Leeds
sctat@leeds.ac.uk
t.a.tarmom@hotmail.com

WILLIAM TEAHAN
School of Computer Science, Bangor University
w.j.teahan@bangor.ac.uk

ERIC ATWELL
School of Computing, University of Leeds
E.S.Atwell@leeds.ac.uk

MOHAMMAD ALSALKA
School of Computing, University of Leeds
M.A.Alsalka@leeds.ac.uk

(*Received November 2018*)

Abstract

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 paper presents detailed research on ways to detect code-switching in Arabic text automatically. We compare the Prediction by Partial Matching (PPM) compression based classifier, implemented in Tawa, and a traditional machine learning classifier Sequential Minimal Optimization (SMO), implemented in Weka, working specifically on Arabic text taken from Facebook. Three experiments were conducted in order to: (1) detect code-switching among 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, Modern Standard Arabic (MSA) and English. Our experiments showed that PPM achieved a higher accuracy rate than SMO with 99.8% v 97.5% in the first experiment and 97.8% v 80.7% in the second. In the third experiment, PPM achieved a lower accuracy rate than SMO with 53.2% v 60.2%. Code-switching between Egyptian Arabic and English text is easiest to detect because Arabic and English are generally written in different character-sets. It is more difficult to distinguish between Arabic dialects and MSA as these use the same character-set, and most users of Arabic, especially Saudis and Egyptians, frequently mix MSA with their dialects. We also note that the MSA corpus used for training the MSA model may not represent MSA Facebook text well, being built from news websites. This paper also describes in detail the new Arabic corpora created for this research and our experiments.

1 Introduction

Code-switching in written natural language text occurs when the author chooses to switch from one language to at least one other. During the last two decades, linguists, sociolinguists and psycholinguists have put forward several definitions for code-switching. Swann and Sinka (2007) believed that the scholar's discipline informed the choice of definitions. Most individuals understand code-switching to occur when mixing between two (or among several) languages. According to Milroy and Muysken (1995), this term is "the alternative use by bilinguals of two or more languages in the same conversation". Myers-Scotton (2006) defined code-switching similarly as "the use of two or more languages in the same dialog". When a bilingual person switches between two languages, this might be due to several reasons or motivations. Some of these are 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.

Code-switching in online communication is prevalent. 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; Androutsopoulos, 2013; Jurgens, Dimitrov and Ruths, 2014; Nguyen et al., 2016). Gupta et al. (2014) stated that users in social media tend to switch from one script, for example Arabic, to another, such as Roman. The occurrence of code-switching in online communication presents a challenge for natural language processing (NLP) tools, since many of them have been prepared for texts written in only one language.

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 and thus make a significant improvement in processing performance.

However, relatively few studies of code-switching for Arabic texts exist, not only on its frequency but also on the development of software to identify occurrences automatically. A contributory factor is that dialect identification for Arabic has been found to be more difficult than for other languages. This paper will seek to address this gap in the research.

The automatic detection of code-switching has been achieved using various approaches such as n-grams (e.g., Oco et al., 2013; Bacatan et al., 2014), dictionary-based methods. Today, detection rates of up to 99% can be reached, even for small input. The work in this paper uses the Tawa toolkit (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 traditional machine learning classifier algorithm which is the Sequential Minimal Optimization (SMO) and the PPM compression-based approach. This paper explains in more detail about creating two types of Arabic corpora, code-switching and non-code-switching, for use in training and testing. Next, it outlines the experiments performed on Arabic Facebook text to evaluate the PPM classifier produced by the Tawa toolkit and the SMO classifier provided by Weka. Finally, it provides a conclusion for this study.

2 Related Work

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

Oco and Roxas (2012) developed pattern-matching refinements (PMRs) to the automatic detection of code-switching by using a dictionary-based approach. They achieved a high accuracy of 94.51%, a marked improvement in accuracy rates over the other dictionary-based approaches, which were in the range of 75.22%–76.26%. The disadvantages of a dictionary-based approach are that dictionaries for some languages may not be available and that some words are not in the dictionaries (Oco et al., 2013).

Lignos and Marcus (2013) produced a system that outlined the problems of both 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. They then annotated 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 96.9% word-level accuracy.

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) and addressed the automatic detection of code-switching in Arabic online text by identifying token-level dialectal words. They mentioned that identifying code-switching in written text is a very challenging task, since an accompanying speech signal does not exist. They produced a system called AIDA (Automatic Identification of Dialectal Arabic) that comprised 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 et al. (2014) used a Naïve-Bayes classifier provided by the Weka toolkit (Hall et al., 2009) to detect code-switching between the Egyptian Arabic Dialect (EAD) and MSA. They used the code-switching portion from the Arabic Online Commentary Dataset built by Zaidan and Callison-Burch (2011). It obtained an accuracy of 51.9%.

Several studies, such as Malmasi et al. (2015) and Malmasi and Zampieri (2016, 2017), have addressed Arabic dialect identification. Ali (2018) used a character-level Convolutional Neural Network (CNN) approach to classify Arabic dialects; this achieved an F1-score of 57.6%. This result was obtained by using a recurrent layer before the convolution layer. Alshutayri et al. (2016) examined several classifiers provided by Weka to classify Arabic dialects. They pointed out that the SMO algorithm achieved the best accuracy (SMO has also been used in this paper as a traditional machine learning classifier for comparison).

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

3 New Arabic Corpora

Most NLP research for the Arabic language focuses on Modern Standard Arabic; research in Arabic dialects is sparse. To evaluate a compression-based approach and traditional machine learning classifiers for the automatic detection of code-switching, it was necessary to build a code-switching corpus containing samples of Arabic code-switching and non-code-switching corpora. We therefore created the Bangor Arabic–English Code-switching (BAEC) corpus for use as a testing set. The following non-code-switching corpora have also been created for this research to use as training sets:

- Saudi Dialect Corpus (SDC);
- Egyptian Dialect Corpus (EDC).

Table 1 lists the number of words, number of characters and overall size in kilobytes for each of these corpora.

Table 1. *Summary of our corpora produced for this research.*

Corpus name	Number of Words	Number of Characters	Corpus size in KB
BAEC	45,251	446,081	436
SDC	210,396	2,065,867	2,018
EDC	218,149	2,072,165	2,024

3.1 Methodology

Different methods were used to create our new corpora. The Facebook Scraper[1] (FS) system helped us to extract data from Arabic Facebook pages. For SDC, we could not collect enough text from Facebook alone, since 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 (Global Media Insight website, 2019) (see Figure 1). Manual cut-and-paste techniques were used to extract data from Twitter and websites. For non-code-switching 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 (Prophet Mohammad's speech, words and actions), since they are classified as Classical Arabic.

3.1.1 Sampling method

A Judgemental, non-probability type sampling method was chosen to collect our data. The procedure for collecting a sample is based on personal judgement, so the researcher uses his or her own experience and knowledge to select a sample (Doyle, 2011). When we built our corpora, therefore, 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 an Egyptian dialect). This required substantial time and effort, taking approximately three months' work.

3.1.2 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 were not commonly written until recently, whereas MSA has official orthographic standards and conventions. Today, Arabic dialects are often used in social media. Two Saudi university researchers with extensive knowledge in Egyptian dialect verified the quality of 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) and came to an agreement. This dictionary contains all MSA words with their meanings.

[1] The Facebook Scraper (FS) system was developed for this research to extract Facebook data automatically (Tarmom, 2018).

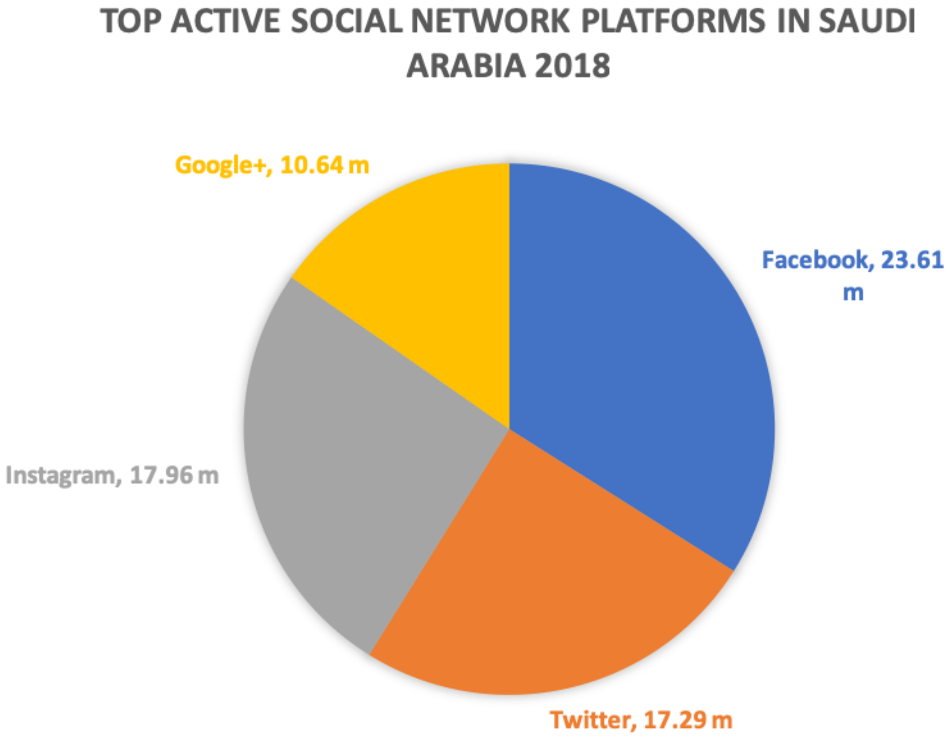


Fig. 1. Most popular social network platforms in Saudi Arabia 2018 (in millions)(Global Media Insight website, 2019).

3.2 Code-Switching Corpus

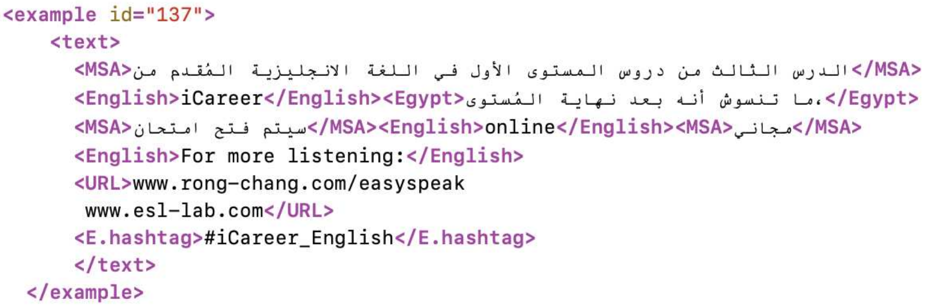
The term "*code-switching corpus*" refers to a body of text consisting of two or more languages under study (Yu et al., 2013). Because Arabic and English are the primary languages in this research, we built an Arabic–English code-switching corpus. To our knowledge, there are no available Arabic code-switching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

The main objective of this research is to detect code-switching in Arabic text from Facebook. We therefore collected our corpus from different Arabic Facebook pages containing code-switching and used it to build the BAEC corpus, which focuses on switching between Arabic and English. It consists of 45,251 words and is 436 KB in size (see Table 1). It was collected from different Facebook pages by using the FS system. 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 2.

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 first thought, as it has 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 write 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>. Also, some users mix Arabic with English words to produce one mixed word, based on a normal word such as 'class', where they add Arabic letters ال and then write الكلاس 'Alclass'. Another habit is 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.

In fact, the detection of TEng and MAE words provided one of the biggest challenges for NLP tools, since the issue was unforeseen and we had insufficient training data to provide an effective means to identify these phenomena.



```
<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>
```

Fig. 2. A sample from the BAEC Corpus.

3.3 Non-Code-Switching Corpora

The term “*non-code-switching corpora*” refers to bodies of text consisting of one language (Yu et al., 2013). Because Arabic is the primary language in this research, we built two Arabic corpora, the Saudi Dialect Corpus (SDC) and the Egyptian Dialect Corpus (EDC) (see Table 1), to be used as non-code-switching corpora for training language models.

3.3.1 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 الخليجيةٓ) spoken in the east region of Saudi Arabia, near the Gulf region, by around 7 million speakers (Simons, Gary and Charles, 2017). The Gulf dialect spreads to Bahrain, Qatar, UAE and Iraq (Simons, Gary and Charles, 2017). In fact, MSA is used throughout all Saudi regions, mixed with each dialect. A map of the dialect usage in Saudi Arabia is shown in Figure 3.

A 210,396-word corpus called the Saudi Dialect Corpus (SDC) was built for training the Saudi model, containing 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 1).

3.3.2 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, Gary and Charles, 2017). In fact, Egyptian TV and cinema spread their dialect to all Arab countries. It is considered the most widely understood dialect in the Arab world. For historical reasons, some frequently-used words are shared between the Egyptian dialect and the Hejazi dialect (one of the Saudi dialects used in the west of Saudi), which makes distinguishing between these two dialects a challenging task.

The Egyptian Dialect Corpus (EDC) that we constructed consists of 218,149 words and is 2,024 KB in size (see Table 1). It was also collected from the social media platform Facebook.

3.4 Analysing the non-code-switching corpus

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 words are used in different corpora and the similarities and differences between them.

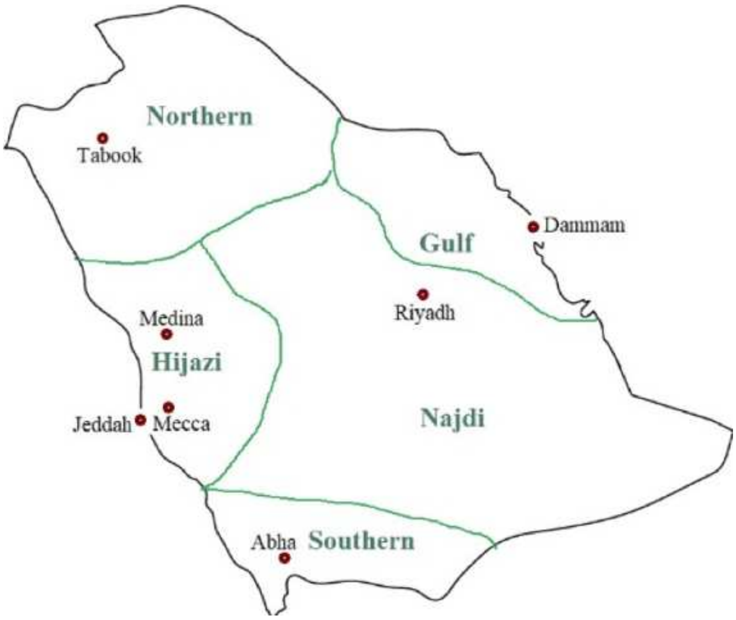


Fig. 3. A map of Saudi Arabia showing the locations of the dialects (based on Al-Moghrabi, 2015).

Table 2 illustrates the top 10 most frequent words from the EDC and the SDC. There are some similarities between the Saudi dialect and the Egyptian dialect. For example, the word من 'of' is the second most frequent word in both dialects. The word فِي 'in' is the most frequent word in the Saudi dialect; in contrast, it is the third most frequent word in the Egyptian dialect.

4 Tawa

Tawa is a compression-based toolkit based on the API designed by Cleary and Teahan (1997); it is an updated version of the text mining toolkit (TMT) (Mahoui et al., 2008). Tawa adopts the Prediction by Partial Matching (PPM) text compression algorithm developed by Cleary and Witten in 1984. It is a character-based model that predicts an upcoming symbol by using the 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 in order to predict the symbol. This is called the escape method and is used to combine the predictions of all character contexts (Cleary and Witten, 1984).

The main aim of the Tawa toolkit is to facilitate the design and implementation of applications that need textual models, such as word/language segmentation, text

Table 2. *The top 10 most frequent words from the EDC and from the SDC.*

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

classification and a wide range of text mining applications, by protecting users from modelling details and estimating process details. It consists of nine main applications, such as `classify`, `codelength`, `train`, `markup`, `segment` and so on (Teahan, 2018). This study concentrates on two applications provided by the Tawa toolkit: building models and language segmentation.

For language segmentation using Tawa, we use multiple models trained on representative text, using the toolkit's `train` tool, from each of the languages under research. We then use the `markup` tool, which utilizes the Viterbi algorithm to find the segmentation with the best compression with all possible segmentation search paths extended at the same time, discarding the poorly performing alternatives (Teahan, 2018).

5 Weka

Weka is a data mining tool that contains different machine learning algorithms for classification, regression, clustering and so on. It has a graphical user interface that makes it easy to use (Hall et al., 2009). Weka implements several classifiers such as Naïve-Bayes, J48, ZeroR, SMO and so on. As mentioned earlier, Alshutayri et al. (2016) pointed out that the SMO algorithm achieved the best accuracy rate when they classified Arabic dialects, so we used the SMO classifier in our experiments.

The Support Vector Machine (SVM) is a supervised machine learning 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 (Li, Bontcheva and Cunningham, 2009). One disadvantage of SVM is that it is slow, especially when applied to a very large classification problem. The enhanced algorithm for SVM, 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).

6 Experiments and Results

Three experiments were performed as part of the evaluation of the compression-based approach (provided by Tawa) and of traditional machine learning classifiers such as the SMO classifier (provided by Weka) to detect code-switching in Arabic Facebook text. These were to: (1) detect code-switching among 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 3.3, the SDC and EDC were created for this purpose. The corpus selected for training MSA was built by Alkahtani, a PhD student at Bangor University (Alkahtani, 2015). The Brown corpus was selected for training English.

The BAEC corpus (described in Section 3.2) was used in these experiments as a testing corpus. First, a cleaning process removed all MAE and TEng words (described in Section 3.2) because we did not have enough training data. Numbers, emojis and punctuation marks were also removed, leaving only pure Arabic and English to process, as we thought this would reduce the errors and enhance the accuracy.

A confusion matrix has been used to evaluate the performance of the automatic detection of code-switching. A confusion matrix is a table that summarizes the classification and segmentation performance. The often-used case is a two-class confusion matrix, used to present the positive and negative classes for some binary classification problems. In this case, the four cells of the matrix are true positives (*TP*), false positives (*FP*), true negatives (*TN*) and false negatives (*FN*), as shown in Table 3 (Sammut and Webb, 2017). *TP* is the number of correct predictions that are positive, *FN* is the number of incorrect predictions that are negative, *FP*

Table 3. *Confusion matrix for two classes.*

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

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 as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

6.1 First experiment

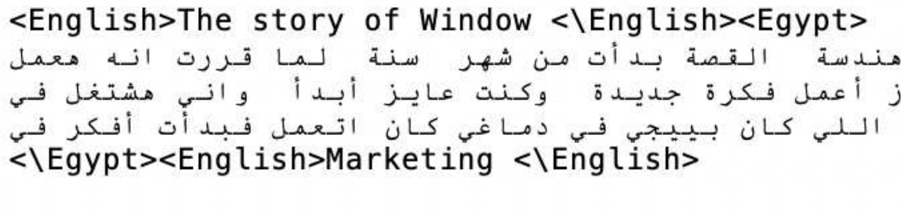
The first experiment was conducted to evaluate the PPM compression algorithm and focused on the detection of code-switching between the Egyptian dialect and English. The testing text was manually extracted from the BAEC corpus for cases that contained only the Egyptian dialect and English, around 17,761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

Automatic detection of code-switching between the Egyptian dialect and English, using the PPM compression algorithm, obtained an accuracy of 99.8%, an average recall of 0.996, an average precision of 0.999 and an average F-measure of 0.998. Table 4 shows the result of this experiment. Some sample output from the PPM classifier is shown in Figure 4.

Table 4. *The results of the first experiment.*

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

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



```
<English>The story of Window <\English><Egypt>
هندسة القصة بدأت من شهر سنة لما قررت انه هعمل
ز أعمل فكرة جديدة وكنت عايز أبدأ واني هشتغل في
اللي كان بييجي في دماغي كان اتعمل فبدأت أفكر في
<\Egypt><English>Marketing <\English>
```

Fig. 4. Sample output from the first experiment's output using PPM.

We repeated this experiment using Weka so we could discover the best result. We used the SMO algorithm with the `StringToWordVector` filter and the `WordTokenizer` filter to detect code-switching between the Egyptian dialect and English. The `WordTokenizer` filter divides the text into words. We achieved an accuracy of 55%, an average recall of 0.55, an average precision of 0.671 and an average F-measure of 0.44. Table 4 shows the result of this experiment.

After that, we examined the SMO 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 shows that UniGram achieved an accuracy of 97.5% which is a higher accuracy rate than the other n-grams models.

6.2 Second experiment

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

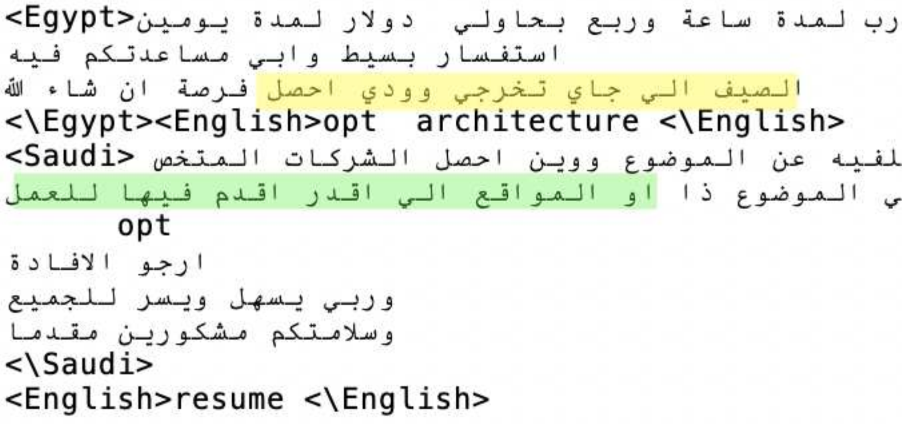
Using PPM in this experiment produced an accuracy of 97.8%, an average recall of 0.899, an average precision of 0.977 and an average F-measure of 0.932. Table 5 shows the result of this experiment.

Figure 5 illustrates some Saudi dialect predicted as Egyptian dialect, such as الصيف الي جاي تخرجي وودي احصل فرصة (highlighted in yellow in Figure 5), and some Saudi dialect correctly predicted as Saudi dialect, such as او المواقع الي اقدر اقدم فيها للعمل (highlighted in green in Figure 5) . Also, the word ‘opt’, an abbreviation in architecture, was predicted as a Saudi dialect word. In this testing text, several

Table 5. *The results of the second experiment.*

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

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 possible abbreviations.



```
<Egypt>رب لمدة ساعة وربع بحاولي  دولار لمدة يومين
استفسار بسيط وابي مساعدتكم فيه
الصيف الي جاي تخرجي وودي احصل فرصة ان شاء الله
<\Egypt><English>opt  architecture <\English>
<Saudi> لفيه عن الموضوع ووين احصل الشركات المتخص
ي الموضوع ذا او المواقع الي اقدر اقدم فيها للعمل
     opt
ارجو الافادة
وربي يسهل ويسر للجميع
وسلامتكم مشكورين مقدما
<\Saudi>
<English>resume <\English>
```

Fig. 5. An example of confusion between the Saudi and Egyptian dialects from the second experiment's output using PPM.

Repeating the second experiment, using the SMO classifier with the `StringToWordVector` filter and the `WordTokenizer` filter, produced an accuracy of 48.2%, an average recall of 0.482, an average precision of 0.571 and an average F-measure of 0.378. Table 5 shows the result of this experiment. We also examined the SMO classifier with the `CharacterNGramTokenize` filter and tried different types of n-grams, as shown in Table 5. Table 5 shows that UniGram achieved an accuracy of 80.7% which is a higher accuracy rate than the other n-grams models.

6.3 Third experiment

The third experiment was considered a more complex task for the PPM classifer since it had four different classes: the Egyptian dialect, the Saudi dialect, MSA and English. The testing file, of 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 0.539, an average precision of 0.562 and an average F-measure of 0.551.Table 6 shows the result of this experiment.

Table 6. *The results of the third experiment.*

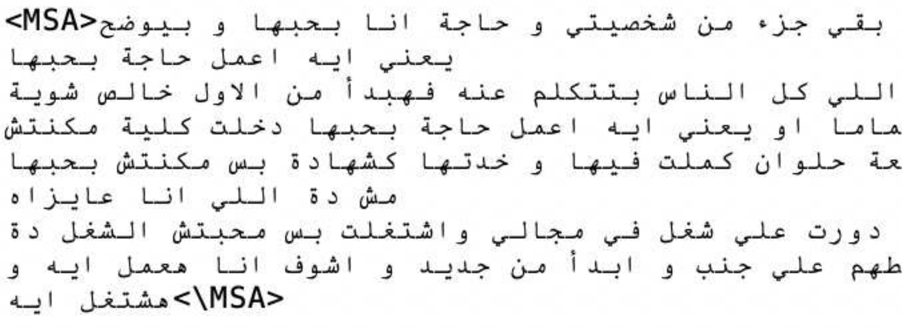
	Accuracy	Precision	Recall	F-Measure
PPM	0.532	0.562	0.539	0.551
SMO (WordTokenizer)	0.263	0.351	0.263	0.239
SMO (UniGram)	**0.602**	**0.680**	**0.602**	**0.597**
SMO (BiGram)	0.511	0.611	0.511	0.421
SMO (TriGram)	0.371	0.553	0.371	0.352
SMO (FourGram)	0.295	0.446	0.294	0.263

Figure 6 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 in Facebook, since 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 7.

Table 7. *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

The Egyptian, Saudi and English models have the lowest minimum code



<MSA>بقي جزء من شخصيتي و حاجة انا بحبها و بيوضح
يعني ايه اعمل حاجة بحبها
اللي كل الناس بتتكلم عنه فهبدأ من الاول خالص شوية
ماما او يعني ايه اعمل حاجة بحبها دخلت كلية مكنتش
عة حلوان كملت فيها و خدتها كشهادة بس مكنتش بحبها
مش دة اللي انا عايزاه
دورت علي شغل في مجالي واشتغلت بس محبتش الشغل دة
طهم علي جنب و ابدأ من جديد و اشوف انا هعمل ايه و
مشتغل ايه<\MSA>

Fig. 6. Sample of some Egyptian dialect sentences predicted as MSA from the third experiment's output using PPM.

lengths with 258722.891 bits, so these are 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.

Repeating the third experiment using the SMO classifier with the `StringToWordVector` filter and the `WordTokenizer` filter obtained an accuracy of 26.3%, an average recall of 0.263, an average precision of 0.351 and an average F-measure of 0.239. Table 6 shows the result of this experiment. We then examined the SMO classifier with the `CharacterNGramTokenize` filter and tried different types of n-grams as shown in Table 6. Table 6 shows that UniGram achieved an accuracy of 60.2% which is a higher accuracy rate than the other n-grams models.

7 Conclusion

This paper discussed our creation of several new Arabic corpora, the production of a code-switching corpus, BAEC, that contains samples of Arabic code-switching and the production of two non-code-switching corpora, SDC and EDC.

We compared the traditional machine learning classifier SMO and the PPM compression-based approach to the automatic detection of code-switching in Arabic text. Our experiments showed, first, that PPM achieves a higher accuracy rate than SMO when the training corpus correctly represents the language or dialect under study. When this condition is satisfied, therefore, 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 SMO, 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 comparison between SMO and PPM, since PPM is a character-based model.

The first experiment focused on the detection of code-switching among the Egyptian dialect and English. PPM obtained an accuracy of 99.8% on testing data from the BAEC corpus, 2.3% higher than the SMO 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%, 17.1% higher than the SMO classifier. Finally, the third experiment detected code-switching among the Egyptian dialect, the Saudi dialect, MSA and English. The SMO classifier obtained an accuracy of 60.2%, 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, since it was built from news websites. As part of future work, a possible solution to overcome this issue is to build a new MSA Facebook corpus, trained on MSA text specially taken from Facebook. In addition, to distinguish 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 improve the results further.

References

Al-Moghrabi, A.A. 2015. *An Examination of Reading Strategies in Arabic (L1) and English (L2) Used by Saudi Female Public High School Adolescents* (Doctoral dissertation, The British University in Dubai (BUiD)). Available at https://bspace.buid.ac.ae/handle/1234/776

Ali, M. 2018. Character level convolutional neural network for German dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)* (pp. 172–177).

Alkahtani, S. 2015. *Building and verifying parallel corpora between Arabic and English* (Doctoral dissertation, Prifysgol Bangor University). Available at http://e.bangor.ac.uk/6546/1/saad_alkahtani_dissertation.pdf

Alkhazi, I.S. & Teahan, W.J. 2017. Classifying and segmenting Classical and Modern Standard Arabic using minimum cross-entropy. *International Journal of Advanced Computer Science and Applications*, 8(4): 421–430.

Alshutayri, A., Atwell, E., Alosaimy, A., Dickins, J., Ingleby, M. & Watson, J. 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)* (pp. 204–211).

Androutsopoulos, J. 2013. Code-switching in computer-mediated communication. In Herring, S. C., Stein, D. & Virtanen, T. (eds.), *Pragmatics of computer-mediated communication*. Berlin, Germany & New York, NY: Mouton de Gruyter. pp. 659–686.

Bacatan, A.C.R., Castillo, B.L.D., Majan, M.J.T., Palermo, V.F. & Sagum, R.A. 2014. Detection of intra-sentential code-switching points using word bigram and unigram frequency count. *International Journal of Computer and Communication Engineering* 3(3): 184.

Cleary, J.G. & Teahan, W.J. 1997. Unbounded length contexts for PPM. *The Computer Journal* 40(2/3): 67–75.

Cleary, J. & Witten, I. 1984. Data compression using adaptive coding and partial string matching.*IEEE transactions on Communications*32(4), pp. 396–402.

Doyle, C. 2011. *A dictionary of marketing.* Oxford, England, UK: Oxford University Press.

Elfardy, H., Al-Badrashiny, M. & Diab, M. 2014. A hybrid system for code switch point detection in informal Arabic text. *XRDS: Crossroads, The ACM Magazine for Students* 21(1): 52–57.

Elfardy, H. & Diab, M. 2012. Token level identification of linguistic code switching. *Proceedings of the International Conference on Computational Linguistics* (COLING): Posters. pp. 287–296.

Global Media Insight website. 2019. Saudi Arabia Social Media Statistics 2018 – Official GMI Blog. [online] *Global Media Insight.* Available at https://www.globalmediainsight.com/blog/saudi-arabia-social-media-statistics/ [Accessed 21 Jun. 2019].

Grosjean, F. 1982. *Life with two languages: An introduction to bilingualism.* Cambridge, UK: Harvard University Press.

Gupta, P., Bali, K., Banchs, R.E., Choudhury, M. & Rosso, P. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval.* ACM, pp. 677–686.

Hale, S.A. 2014. Global connectivity and multilinguals in the Twitter network In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 26 April 2014, Toronto, Canada.* ACM, pp. 833–842).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11(1): 10–18.

Johnson, I. 2013. Audience design and communication accommodation theory: use of Twitter by Welsh–English biliterates. In Jones, E.H.G. & Uribe-Jongbloed, E. (eds.) *Social Media and Minority Languages: convergence and the creative industries.* Bristol: Multilingual Matters, pp. 99–118.

Jurgens, D., Dimitrov, S. & Ruths, D. 2014. Twitter users# codeswitch hashtags# moltoimportante# wow. In *Proceedings of the First Workshop on Computational Approaches to Code-switching, 25 October 2014, Doha, Qatar* (pp. 51–61).

Li, Y., Bontcheva, K. & Cunningham, H. 2009. Adapting SVM for data sparseness and imbalance: a case study in information extraction. *Natural Language Engineering* 15(2): 241–271.

Lignos, C. & Marcus, M. 2013. Toward web-scale analysis of codeswitching. In *87th Annual Meeting of the Linguistic Society of America, 3 January 2013, Boston.*

Malmasi, S., Refaee, E. & Dras, M. 2015. Arabic dialect identification using a parallel multidialectal corpus. In *Conference of the Pacific Association for Computational Linguistics.* Singapore: Springer, pp. 35–53.

Malmasi, S. & Zampieri, M. 2016. Arabic dialect identification in speech transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)* (pp. 106–113).

Malmasi, S. & Zampieri, M. 2017. Arabic dialect identification using iVectors and ASR transcripts. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)* (pp. 178–183).

Mahoui, M., Teahan, W.J., Thirumalaiswamy Sekhar, A.K. & Chilukuri, S. 2008. Identification of gene function using prediction by partial matching (PPM) language models. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management.* ACM, pp. 779–786.

Milroy, L. & Muysken, P. (eds.). 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching.* Cambridge University Press.

Myers-Scotton, C. 2006. Multiple voices: An introduction to bilingualism. London: Blackwell.

Nguyen, D., Doğruöz, A.S., Rosé, C.P. & de Jong, F. 2016. Computational sociolinguistics: A survey. *Computational linguistics.* Available at https://arXiv:1508.07544v2

Oco, N. & Roxas, R.E. 2012. Pattern matching refinements to dictionary-based code-switching point detection. In *Pacific Asia Conference on Language, Information and Computation (PACLIC), 7 November 2012, Bali, Indonesia* (pp. 229–236).

Oco, N., Wong, J., Ilao, J. & Roxas, R. 2013. Detecting code-switches using word bigram frequency count. In *9th National Natural Language Processing Research Symposium, Quezon City, Philippines, March* (Vol. 7).

Platt, J. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. In *Technical Report MSR-TR-98-14.* Microsoft Research.

Sammut, C. & Webb, G.I. (2017). *Encyclopedia of machine learning and data mining.* US: Springer.

Simons, G.F. & Fennig, C.D. (eds.) 2017. Ethnologue: Languages of the world, twentieth edition. Dallas, Texas: SIL International. *Online version: http://www.ethnologue.com.*

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A. & Fung, P. 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, 25 October 2014, Doha, Qatar* (pp. 62–72).

Swann, J. & Sinka, I. 2007. Style shifting, code switching. In: Graddol, D., Leith, D., Swann, J., Rhys, M. & Gillen, J. (eds.) *Changing English.* London, UK: Routledge.

Tarmom, T. 2018.*Designing and Evaluating a Compression-based Approach to the Automatic Detection of Code-switching in Arabic Text* (MSc dissertation, Bangor University).

Teahan, W. 2018. A Compression-Based Toolkit for Modelling and Processing Natural Language Text. *Information* 9(12): 294.

Yu, L.C., He, W.C., Chien, W.N. & Tseng, Y.H. 2013. Identification of code-switched sentences and words using language modeling approaches. *Mathematical Problems in Engineering*, 2013, Article ID 898714.

Zaidan, O.F. & Callison-Burch, C. 2011. 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.* Association for Computational Linguistics, pp. 37–41.