Compression-Based Parts-of-Speech Tagger for The Arabic Language

Ibrahim S Alkhazi *i.alkhazi@ut.edu.sa*
College of Computers & Information Technology
Tabuk University
Tabuk, Saudi Arabia

William J. Teahan *w.j.teahan@bangor.ac.uk*
School of Computer Science and Electronic Engineering
Bangor University
United Kingdom

Abstract

This paper explores the use of Compression-based models to train a Part-of-Speech (POS) tagger for the Arabic language. The newly developed tagger is based on the Prediction-by-Partial Matching (PPM) compression system, which has already been employed successfully in several NLP tasks. Several models were trained for the new tagger, the first models were trained using a silver-standard data from two different POS Arabic taggers, and the second model utilised the BAAC corpus, which is a 50K term manually annotated MSA corpus, where the PPM tagger achieved an accuracy of 93.07%. Also, the tag-based models were utilised to evaluate the performance of the new tagger by first tagging different Classical Arabic corpora and Modern Standard Arabic corpora then compressing the text using tag-based compression models. The results show that the use of silver-standard models has led to a reduction in the quality of the tag-based compression by an average of 0.43%, whereas the use of the gold-standard model has increased the tag-based compression quality by an average of 4.61% when used to tag Modern Standard Arabic text.

Keywords: Natural Language Processing, Arabic Part-of-Speech Tagger, Hidden Markov Model, Statistical Language Model.

1. BACKGROUND AND MOTIVATION

A parts-of-speech (POS) tagger is a computer system that accepts text as input and then assigns a grammatical tag, such as VB for a verb, JJ for an adjective and NN for a noun, as output for every token or term according to its appearance, position or order in the text. POS tagging is normally the initial step in any linguistic analysis and a very significant early step in the process of building several natural language processing (NLP) applications, such as information retrieval systems as shown in Figure 1, spell auto-checking, correction systems and speech recognition systems (Abumalloh et al. 2016). Alabbas and Ramsay (Alabbas and Ramsay 2012) argue that higher tagging accuracy improves the quality of all subsequent stages and therefore, assessing the tagger accuracy is an important step in the development of many NLP tasks.

The tagging process can be achieved by one of the following general methods: (1) a statistical approach where a language model is trained using previously tagged corpora, such as the BAAC (Alkhazi and Teahan 2018) and the Arabic Treebank (Hajic et al. 2004), and the model is then used to tag different text; (2) a rule-based approach where linguists define and develop rules or knowledge base, as shown in Figure 2, which are used to assign POS tags; and (3) by combining the previous two approaches in a hybrid system (Alosaimy 2018; Atwell, Elsheikh, and Elsheikh 2018; El Hadj, Al-Sughayeir, and Al-Ansari 2009; Khoja 2003; Al Shamsi and Guessoum 2006).

The earliest approach used for developing POS taggers is the rule-based method (Abumalloh et al. 2016; Khoja 2001, 2003), that was first developed in the 1960s. As stated before, this method utilises a collection of linguistic rules, where the number of rules ranges from hundreds to thousands, to tag the text. The development of a rule-based tagger is difficult, costly and the system is usually not quite robust (Abumalloh et al. 2016). Brill (Brill 1992) developed the TBL rule-base tagger that obtained a tagging accuracy similar to that of statistical taggers. Unlike statistical taggers, the linguistic knowledge is created automatically as Brill's tagger trains simple non-stochastic rules (Brill 1992). Other examples of rule-base taggers are the CGC tagger developed by Klein and Simmons (Klein and Simmons 1963), the TAGGIT tagger which was produced by Greene and Rubin (Greene and Rubin 1971). Nguyen and others have developed a rule-based POS tagger that utilises an SCRDR tree (Richards 2009), as shown in Figure 2, to represent the rules used by the RDRPOSTagger (Nguyen et al. 2014). RDRPOSTagger was utilised to tag two languages, English and Vietnamese, with a reported accuracy of 93.51%. The tagger uses an error-driven procedure to build the knowledge base automatically in the form of a binary tree as shown in Figure 2.

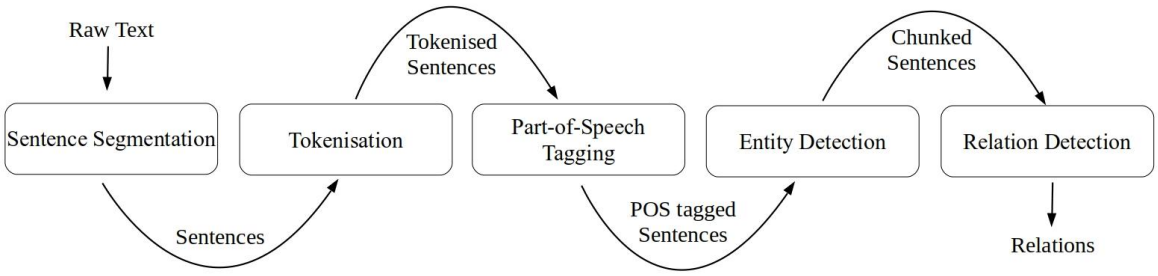

FIGURE 1: Simple information retrieval system pipeline architecture (nltk.org n.d.).

In 1990, the statistical approach started to substitute the rule-based POS tagging approach, and according to Martinez (Martinez 2012), the statistical approach also started to be adopted more with several other NLP tasks, reporting state-of-the-art results. The Markov modelling method (as applied in PPM models) has been successfully applied to many areas of NLP. PPM language modelling achieves state-of-the-art compression of the text written in many languages, with results reported in (Alhawiti 2014; Alkhazi, Alghamdi, and Teahan 2017; Teahan 1998). Another NLP application of PPM involves word segmentation of Chinese text, in this case by adding spaces to Chinese text that has no spaces (Teahan et al. 2000). Many other NLP tasks in other languages, such as code switching, authorship attribution, text correction, cryptology and speech recognition, were reported in various studies such as (Al-Kazaz, Irvine, and Teahan 2016; Alghamdi, Alkhazi, and Teahan 2016; Alhawiti 2014; Alkahtani and Teahan 2016; Alkhazi and Teahan 2017; Teahan 1998, 2000).

Prediction by Partial Matching (PPM) is an online adaptive text compression system that utilises the prior context to predict the coming symbol or character with given fixed context length. It utilises a Markov-based n-gram method with a backing-off mechanism similar to a method which Katz (Katz 1987) proposed in 1987. Nonetheless, PPM introduced the “escaping” mechanism prior to Katz’s suggested method. In 1984, Cleary and Witten (Cleary, John and Witten 1984) were first to introduce the system when they proposed the two PPM character-based variants, PPMA and PPMB. Later, in 1990 and 1993, two more modifications of PPM, PPMC and PPMD (Wu 2007), were introduced by Moffat and Howard. The main difference between these modifications of PPM, PPMA, PPMB, PPMC and PPMD, is how they estimate the escape probability which is the smoothing technique needed by the model to back off to a decreased order. Many experiments have revealed that text compression using PPMD normally gives better compression compared to other variants of PPM (Khmelev and Teahan 2003).

Text compression can be achieved in three main ways using the PPM algorithm. The first way is the use of character-based models in which the preceding context of observed symbols or characters is applied to foretell the next one. The other method of applying PPM is to use the word-based modelling of the text in which the trained model utilises the previous context of the observed word to foretell the imminent word. The final method employs tag-based models that utilise the previously foretold tags and words to predict the imminent terms (both tags and words) (Teahan 1998). The concept of the tag-based method, as shown in Figure 3, is that recognising the tag of the term aids in predicting it. The principal advantage of employing the tag to foretell the imminent term is that the tag will in all probability have appeared many more times previously, and consequently be a better foreteller for the forthcoming tags plus terms (Brown et al. 1992; Jelinek 1990; Kuhn and De Mori 1990).

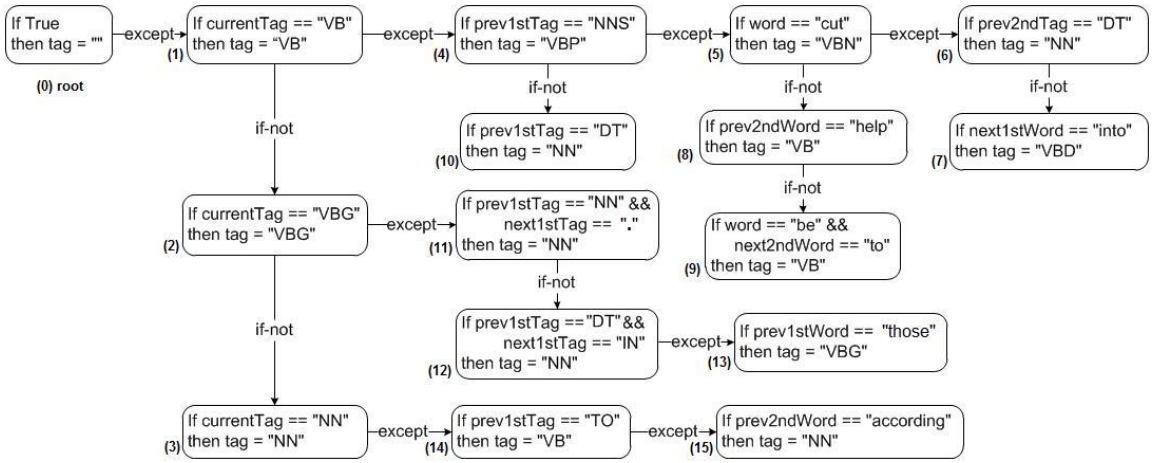

FIGURE 2: A sample of RDRPOSTagger tagging rules (Nguyen et al. 2014).

The tag-based model foretells the imminent term by utilising two streams, a tag and a term stream as shown in Figure 3. In the beginning, the tag-based model will utilise a PPM model that uses the two prior tags to foretell the next one. Next, using the tag along with the earlier observed term, the tag-based model will attempt to foretell the imminent term. If the sequence or its prediction has not been seen before, the model will encode an escape probability and it will attempt to maintain foretelling the next term utilising just the current tag. Finally, if the model's prediction of the current term is unsuccessful, a character-based model will be utilised (Teahan and Cleary 1998). To compress the text using the tag-based model, the text must be tagged first, as both the terms and tags sequences will effectively be encoded together. Depending on the language and tagset used, compressing the tags along with the words can lead to better overall compression despite the cost of encoding the extra tag information (Alkhazi et al. 2017; Teahan et al. 1998; Teahan and Cleary 1998). Further studies and results about tag-based modelling are reported by Teahan and Alkhazi (Alkhazi et al. 2017; Teahan and Cleary 1998).

This research used the Tawa toolkit (Teahan 2018) to perform the tag-based compression of the text. According to Teahan (Teahan 2018), "The aim of the toolkit is to simplify the conceptualisation and implementation for a broad range of text mining and NLP applications involving textual transformations". The toolkit can be used to implement a wide spectrum of NLP applications and it comprises eight principal applications such as `train`, `encode`, `decode` and `classify`. It adopts a 'noiseless channel model' design where every application is conceived as an encoding process without loss of any information and any procedure is reversible. The algorithms and pseudo-code of the encoding, decoding, training and six other applications are described in detail by Teahan (Teahan 2018). Other details, such as the implementation aspects and search algorithms applied in the toolkit, are also addressed by the developer.

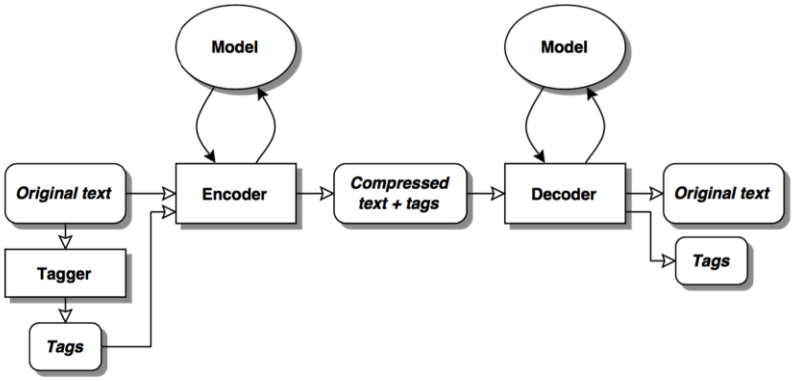

FIGURE 3: Tag-based compression of Arabic text (Teahan and Cleary 1998).

The Arabic language "العربية" is among the most popular languages in use today, as shown in Figure 4. In the United Nations, it is among the five official languages and it is the primary language of 330 million people living in 22 countries in Asia, North Africa and the Middle East along with it being a secondary language of 1.4 billion people (Soudi et al. 2012). Arabic is a morphologically rich language having a mutual structure with Semitic languages such as Tigrinya, Hebrew and Amharic. The Arabic language uses a right-to-left writing system with a verb-subject-object (VSO) grammatical structure. Other structures, such as VOS, VO and SVO are also possible in the language (Al-Harbi et al. 2008; Alghamdi et al. 2016; Green and Manning 2010).

Arabic has a morphological complex natural that causes various difficulties for Natural Language Processing (NLP) (Alosaimy 2018; Columbia University n.d.; Habash, Rambow, and Roth 2009; Pasha et al. 2014). Diacritics are used in the Arabic language to disambiguate terms. The presence of the four diacritics, which are FatHa, Dhamma, Kasra and Sukuun, in the text help in the lexical disambiguation of the word, as some words share identical component letters but different diacritics. Modern Standard Arabic text is very commonly written without diacritics and the contextual information is used by the reader of the text to disambiguate the meaning of the term. As a result of the ambiguity problem, the use of the Rule-based approach to tag the text increases the number of unanalyzed and mistagged terms (Hadni et al. 2013). The statistical method of tagging the Arabic text is broadly utilised to solve the POS uncertainty of the Arabic text (Al Shamsi and Guessoum 2006).

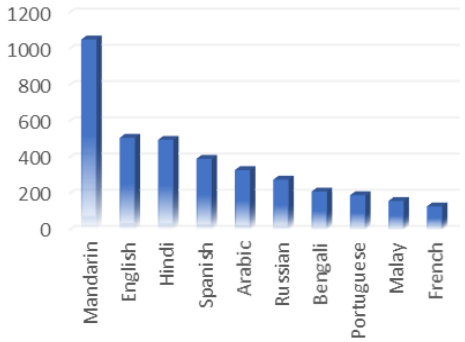

FIGURE 4: The most globally used languages (Alkhazi and Teahan 2017).

The tagset is a list of all the potential tags which could be assigned to the terms while the tagging process and it is regarded as a fundamental component for any POS tagger. For the English language, there are a modest number of common tagsets which are developed and used by English POS taggers. For example, the Brown tagset used in the Brown corpus which comprises

226 tags, the LOB tagset used in the LOB corpus, which is based on the Brown tagset, containing 135 tags (Francis and Kucera 1979), and the Penn Treebank tagset which was used to tag the Penn Treebank corpus and contained 36 tags (Taylor, Marcus, and Santorini 2003).

For the Arabic language, tagsets can be divided into traditional and English derived tagsets (Alosaimy 2018). English derived tagsets arose when Arabic resources were limited, and a tagset is urgently needed to develop new resources (Diab 2007; Hajic et al. 2004; Maamouri and Bies 2004). This type of tagset is usually a trivial modification of the standard English tagset, and this modification was considered problematic for Semitic languages as stated by Wintner (Wintner 2014), and illustrated by Alosaimy who showed that in some cases differentiation among adjectives and nouns is unclear (Alosaimy 2018). Many traditional tagsets for the Arabic language have been proposed. For example, the Khoja tagset utilised by the APT tagger includes 177 tags (Khoja 2001, 2003). The El-Kareh and Al-Ansary (El-Kareh and Al-Ansary 2000) tagset comprises 72 tags used in their tagger. Al-Shamsi and Guessom (Al Shamsi and Guessoum 2006) proposed a tagset that includes 55 tags, which was employed in the HMM tagger that they have developed. Finally, Al-Qrainy (Alqrainy 2008) proposed a new tagset, that was used in AMT tagger that comprises 161 detailed tags and 28 general ones.

The tagset used in this research is the same as used by the Madamira tagger (Pasha et al. 2014), which was used initially by the MADA tagger (Habash et al. 2009). The tagset is the subset of the English tagset which was presented with the English Penn Treebank and consists of 32 tags and was initially proposed by Diab, Hacioglu and Jurafsky (Diab, Hacioglu, and Jurafsky 2004). The experiments conducted by Alkhazi, Alghamdi and Teahan (Alkhazi et al. 2017) have concluded that the quality of tag-based compression varies from one tagset to another. The different tagsets, some of which are shown in Table 1, were used to compress MSA text using POS tags, and tag-based compression using the Madamira tagset outperforms other tagsets such as Stanford (Green, de Marneffe, and Manning 2013) and Farasa (Abdelali et al. 2016). Since the main goal of this research is to investigate the use of the PPM compression scheme to develop and train a new Arabic POS tagger, and based on the results concluded by Alkhazi and Teahan (Alkhazi et al. 2017; Alkhazi and Teahan 2018), which stats that Madamira tagger tag-based compression results outperformed other taggers, the Madamira tagset and tagger output will be adopted in this research.

The work in this paper uses an approach based on the Prediction-by-Partial Matching (PPM) compression scheme to develop and train a new Arabic POS tagger. This Markov-based approach effectively has been employed in many NLP tasks in the past often with state-of-the-art results or results competitive with traditional schemes (Al-Kazaz et al. 2016; Alghamdi et al. 2016; Alkhazi et al. 2017; Teahan 1998, 2000; Teahan et al. 2000; Teahan and Cleary 1997). It will first discuss the two parts of the experiment, where silver-standard data is used in the first section to train the Tawa Arabic POS Tagger (TAPT), and a gold-standard data, the BAAC corpus, is used in the second section as a training data. Secondly, the BAAC will be used to evaluate the tagger and limitations of those experiments are discussed in detail. In both sections, the effectiveness of using silver and gold-standard models will be examined by utilising the tag-based models to compress CA and MSA corpora tagged by the TAPT. Finally, the conclusion and future work are presented.

2. EXISTING STATISTICAL ARABIC POS TAGGERS

The Madamira tagger is a disambiguation and morphological analysis system which can perform various natural language processing tasks for the Arabic language such as tokenization, part-of-speech tagging, phrase chunking and other tasks (Pasha et al. 2014). According to Pasha and others (Pasha et al. 2014), Madamira blends and improves some of the best services that the previously two used systems, MADA (Habash et al. 2013, 2009; Habash and Rambow 2005) and AMIRA (Diab, Hacioglu, and Jurafsky 2007), provide. The system was trained using the first three parts of the Penn Arabic Treebank, ATC. It supports both XML and plain text as input and output file type, and an online demo (Pasha et al. 2014) of MADAMIRA is made available at (Anon n.d.). The Madamira tagset used in this paper consists of 32 tags. There are several steps in

Madamira's preprocessor of the text. First, it transliterates the text using the Buckwalter transliterator (Tim Buckwalter n.d.). Then, it utilises the SAMA and CALIMA Analysers to morphologically analyse the text. Next, it creates SVM language models. Then, Madamira uses the morphological features to tokenise the text. The final step is performing the phrase chunking and named entity recognition of the text by utilising SVM models (Atwell et al. 2018).

The Stanford Arabic tagger is a Support Vector Machine (SVM) based tagger developed at Stanford University. It is an open-sourced, multi-language, Java-based tagger that utilises a maximum entropy modelling technique, which according to Green, Marneffe and Manning (Green et al. 2013) can achieve a tagging accuracy of 95.49% (Atwell et al. 2018). The Stanford tagger was trained to tag other languages such as German, Spanish, French and Chinese and provides a command-line interface and an API. The first three parts of the Arabic Penn Treebank were used to train the Stanford Arabic tagger (Anon n.d.). The tagset of this tagger consists of 24 tags. Those tags are derived by manually decreasing the 135 tags obtained from the Arabic Treebank distribution (Alkhazi et al. 2017; Atwell et al. 2018).

In 2002, the APT Arabic tagger was developed by Khoja (Khoja 2003; Khoja, Garside, and Knowles 2001). The tagger uses the hybrid approach with a tagset that is based on the BNC English tagset and consists of 131 tags. According to the author, the tagger reached an accuracy of 86%. Mohamed and Kübler (Mohamed and Kübler 2010) have developed an Arabic POS tagger that utilises two approaches, the first requires no segmentation of the word and the second applies the basic POS word segmentation. According to Mohamed and Kübler, the first approach achieved an accuracy of 93.93% and the second approach achieved an accuracy of 93.41%. Al Shamsi and Guessoum (Al Shamsi and Guessoum 2006) used a statistical method which employs HMMs to train an Arabic POS tagger. The tagger, which utilises Buckwalter's stemmer and uses a tagset that includes 55 tags, achieved an accuracy of 97%. Darwish and others (Darwish et al. 2018) have developed a POS tagger that tags four different Arabic dialects, which are Gulf, Maghrebi, Egyptian and Levantine. The tagger, which was trained by a new dataset that contains Arabic tagged tweets, has achieved an accuracy of 89.3%.

Term	Madamira Tag	Stanford Tag	Farasa Tag
وقد	part_verb	NN	PART
اتخذت	verb	VBD	V
خطوات	noun	NNS	NOUN-FP
بإنشاء	noun	NN	NOUN-MS
لجنة	noun	NN	NOUN-FS
الحقيقة	noun	DTNN	NOUN-FS
والمصالحة	noun	NN	NOUN-FS
واللجنة	noun	NN	NOUN-FS
الوطنية	adj	DTNN	ADJ-FS
المستقلة	adj	DTJJ	ADJ-FP
لحقوق	noun	NN	NOUN-FS
الإنسان	noun	DTNN	NOUN-MS

TABLE 1: Sample of various Arabic tagsets.

3. EXPERIMENTS AND RESULTS

3.1 Data Source

In the first section of the experiments, two sub-corpora of Corpus A (Alkahtani and Teahan 2016) were used to train the TAPT. Corpus A is an MSA corpus that includes various topics such as politics, opinions, legal issues, economics, conferences, business, cinema and books. The text in the corpus was gathered from the Al-Hayat website, a bilingual newspaper, and from the open-

source online corpus, OPUS (Alkahtani 2015). The second section of the experiments has utilised the BAAC corpus to train and evaluate TAPT. The Bangor Arabic Annotated Corpus (BAAC) (Alkhazi and Teahan 2018) is an MSA corpus that comprises 50K words manually annotated by parts-of-speech. The data source for the new corpus is the Press sub-corpus from the BACC corpus (Alhawiti 2014), which was created originally to test the performance of various text compression algorithms on different text files. The results of the text classification performed by Alkhazi and Teahan (Alkhazi and Teahan 2017) revealed that the Press sub-corpus is 99% written in MSA, as shown in Figure 5. According to the authors, the sub-corpus is a newswire text consisting of 50K terms, gathered from various news websites between 2010 and 2012 and covers many topics such as political and technology news.

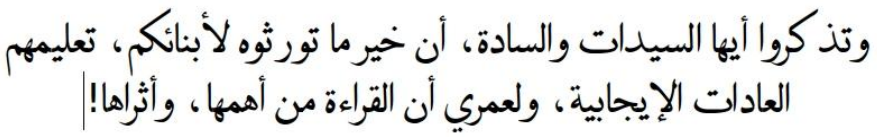

FIGURE 5: Sample text from the Press sub-corpus (Alhawiti 2014).

A new one-to-one transliteration tool was developed and then used in both experiments to transliterate Arabic characters to Latin characters. The new tool is based on the Buckwalter Arabic transliteration tool (Linguistic Data Consortium. 2002; Tim Buckwalter n.d.) developed by Tim Buckwalter. The new mapping, as shown in Table 2, adds Arabic numbers and some Quranic symbols that were found in CA corpora used in the experiments. The tool was utilised to transliterate training and input text for the TAPT to Latin characters and the output tagged text to Arabic characters.

Arabic Character	Latin Character	Arabic Character	Latin Character	Arabic Character	Latin Character
\u0621	q	\u0634	z	\u064C	D
\u0622	w	\u0635	x	\u064D	F
\u0623	e	\u0636	c	\u064E	R
\u0624	r	\u0637	v	\u064F	W
\u0625	t	\u0638	b	\u0650	U
\u0626	y	\u0639	n	\u0651	S
\u0627	u	\u063A	m	\u0652	E

TABLE 2: A sample of the new character mapping.

3.2 Silver-standard Data Experiment

This section illustrates the use of silver-standard data, which was tokenised and tagged using both the Madamira and the Stanford taggers, to train and then evaluate the TAPT. The experiment was conducted as follows:

- Corpus A was first tokenised then tagged using Madamira and the Stanford taggers.
- Then, the text was preprocessed and input into the Tawa toolkit (Teahan 2018) then transliterated to Latin characters.
- Next, two PPM tagging models were created, the first model was trained using Madamira tagged text and the second model was trained using Stanford tagged text.
- Finally, a smaller version of the BAAC corpus, that has only 5K terms, was selected then tagged using the two models from the previous step.

To calculate the accuracy of using silver-standard data to train the TAPT, the Madamira and Stanford gold-standard data utilised by Alkhazi and Teahan (Alkhazi and Teahan 2018) was used to establish the number of incorrectly assigned tags. The tagger achieved an accuracy of 84.37%, with 794 incorrectly assigned tags, using the Madamira silver- standard model, and 81.75% using the Stanford silver-standard model with 927 incorrectly assigned tags. Table 3 shows the most incorrectly assigned tags for the TAPT which was trained by silver-standard text tagged by Madamira and Stanford POS taggers.

Frequency	Madamira Assigned Tag	BAAC Tag	Frequency	Stanford Assigned Tag	BAAC Tag
165	noun	verb	118	JJ	DTJJ
51	noun	adj	64	NN	NNP
46	conj_sub	verb_pseudo	48	VBD	VBP
34	noun	abbrev	45	VBD	NN
27	adj	noun	44	RP	NN
21	noun_prop	noun	37	NNP	NN
20	prep	verb_pseudo	37	NN	JJ
17	verb	abbrev	36	NNP	DTNN
17	noun	noun_prop	24	DTNNS	DTNN
16	prep	part_neg	22	RB	NN

TABLE 3: Top 10 most incorrectly assigned tags for the TAPT trained on silver-standard Madamira and Stanford models.

The results in Table 3 show that almost 25.56% of the incorrectly assigned tags by the TAPT that used the Madamira model were in fact verbs and 8.18% were nouns, which includes noun_prop and noun. Compared to the Stanford model, only 5.17% of the inaccurately assigned tags by the TAPT that used the Stanford model were in fact verbs whereas 29.34% of the inaccurately assigned tags were nouns, that includes NNP, NN and DTNN. The previous results confirm the results reported by Alkhazi and Teahan (Alkhazi and Teahan 2018) which suggest that there is an issue in the process of assigning the verb tag by the Madamira tagger and the noun tag by the Stanford tagger.

To evaluate the performance of the TAPT that was trained on Madamira silver-standard text, the BACC corpus (Alhawiti 2014) was tagged then compressed using tag-based compression models. The BACC corpus as stated by Alkhazi and Teahan (Alkhazi and Teahan 2018), is a mixture of MSA and CA text. Table 4 and Table 5 represent the results of compressing the BACC sub-corpora 'Arabic History', 'Arabic Literature', 'Art and Music' and 'Sports'. The two tables show that the tag-based compression performance on the text that was tagged by the TAPT, that was trained on silver-standard text, has decreased compared to the performance of the Madamira tag-based compression.

Sub-text	Text Type	Corpus Size	Character-based Compression size	Madamira Tag-based Compression size	TAPT Tag-based Compression size
Arabic History	CA	30251137	4206076	4267257	4290052
Arabic Literature	CA	18594383	3029433	3045281	3067010
Art and Music	MSA	41770	9510	10583	10604
Sports	MSA	31059	6497	7124	7149

TABLE 4: The character-based and the tag-based compression results of the Madamira and the TAPT trained on silver-standard corpus.

Sub-text	Text Type	Character-based bpc	Madamira bpc	TAPT bpc	TAPT Performance Decrease
Arabic History	CA	1.11	1.13	1.13	-0.52%
Arabic Literature	CA	1.3	1.31	1.32	-0.70%
Art and Music	MSA	1.82	2.03	2.03	-0.18%
Sports	MSA	1.67	1.83	1.84	-0.32%

TABLE 5: The decrease in the tag-based compression performance of TAPT trained on silver-standard text compared to the Madamira tagger.

3.3 Gold-standard Data Experiment

This section represents the use of a gold-standard annotated text, the BAAC corpus, to train and then evaluate TAPT. Using a tenfold cross validation method, TAPT achieved an accuracy of 93.07% when trained using the BAAC corpus. Table 6 shows the most frequently assigned tags by TAPT and Table 7 displays the most incorrectly assigned tags compared to the tag at the BAAC corpus.

Frequency	Tag
24787	noun
5693	prep
5584	verb
4431	adj
2519	noun_prop
1656	conj_sub
1148	conj
985	pron_rel
765	pron_dem
599	noun_quant
500	part_neg
355	pron
329	adv
251	noun_num

TABLE 6: The most frequently assigned tags by the TAPT trained on gold-standard text.

To evaluate the performance of the TAPT when trained on gold-standard text, four BACC sub-corpora were first tagged by the TAPT and then the text was compressed using tag-based compression models. Table 8 compares the results of compressing the BACC sub-corpora 'Arabic History', 'Arabic Literature', 'Art and Music' and 'Sports' using the character-based and the tag-based model. Both 'Arabic History' and 'Arabic Literature' are 99% written in CA text, whereas 'Art and Music' and 'Sports' are 91% and 95% consecutively, written in MSA text. Table 9 shows the tag-based compression ratio (in bits per character) of the four BACC sub-corpora which were tagged by the TAPT and the Madamira tagger. It is noticeable that the quality of compression of the 'Art and Music' and 'Sports' sub-corpora has increased by 4.98% and 4.25% respectively, whereas the compression quality of the sub-corpora, 'Arabic History' and 'Arabic Literature', has decreased by 2.69% and 1.56% respectively, compared to the tag-based compression results of the Madamira tagger.

The results in Table 8 and 9 indicate that tagging MSA text using the TAPT increases the quality of the tag-based compression compared to the Madamira tagged text. The results also show that the quality of the tag-based compression of CA text that was tagged by the TAPT has decreased. A possible cause of improvement in compressing the MSA corpora is the fact that the TAPT is trained using the BAAC corpus which according to Alkhazi and Teahan [23], is 99% written in MSA.

Frequency	PPM Assigned Tag	BAAC Tag
73	noun	adj
45	adj	noun
41	verb	noun
19	noun	verb
12	noun_prop	noun
12	noun	conj
11	noun	noun_prop
10	conj_sub	verb_pseudo
5	noun_prop	verb
5	adv	adv_interrog

TABLE 7: Top 10 most incorrectly assigned tags for the TAPT trained on gold-standard corpus.

4. CONCLUSION AND FUTURE WORK

This paper presented a newly developed compression-based POS tagger for the Arabic language which is based on a Prediction-by-Partial Matching (PPM) compression system. The results of the tagger were presented in two experiments. The first used models which were trained using silver-standard data from two different POS Arabic taggers, the Stanford and the Madamira taggers (Columbia University n.d.; Pasha et al. 2014). The results of the previous experiment show that using silver-standard data to train the TAPT decreases the quality of the tag-based compression of both the CA and MSA text compared to the Madamira tagger. The second experiment trained a model using the BAAC corpus, which is a 50K term manually annotated MSA corpus, where the TAPT achieved an accuracy of 93.07%. The tag-based compression results of the second experiment show that the use of the gold-standard model increases the quality of the tag-based compression when the TAPT is used to tag MSA text.

Sub-text	Text Type	Corpus Size	Character-based Compression size	Madamira Tag-based Compression size	TAPT Tag-based Compression size
Arabic History	CA	30251137	4206076	4267257	4387191
Arabic Literature	CA	18594383	3029433	3045281	3093824
Art and Music	MSA	41770	9510	10583	10027
Sports	MSA	31059	6497	7124	6807

TABLE 8: The character-based and the tag-based compression results of the Madamira and the TAPT trained on gold-standard corpus.

Future enhancements to the tagger can be made by utilising more Arabic resources, such as the 'Sunnah Arabic Corpus' (Alosaimy 2018) which is a set of CA text that is popularly cited in Islamic books and the ATB corpus (Hajic et al. 2004). Including such resources might increase the accuracy of the TAPT.

Sub-text	Text Type	Character-based bpc	Madamira bpc	TAPT bpc	TAPT Improvement
Arabic History	CA	1.11	1.13	1.16	-2.69%
Arabic Literature	CA	1.30	1.31	1.33	-1.56%
Art and Music	MSA	1.82	2.03	1.92	4.98%
Sports	MSA	1.67	1.83	1.75	4.25%

TABLE 9: The Tag-based compression improvement of TAPT trained on gold-standard corpus compared to the Madamira tagger.

5. REFERENCES

[1] Abdelali, Ahmed, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. "Farasa: A Fast and Furious Segmenter for Arabic." Pp. 11–16 in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.

[2] Abumalloh, Rabab Ali, Hassan Maudi Al-Sarhan, Othman Ibrahim, and Waheeb Abu-Ulbeh. 2016. "Arabic Part-of-Speech Tagging." Journal of Soft Computing and Decision Support Systems 3(2):45–52.

[3] Al-Harbi, S., A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh. 2008. "Automatic Arabic Text Classification." in Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data.

[4] Al-Kazaz, Noor R., Sean A. Irvine, and William J. Teahan. 2016. "An Automatic Cryptanalysis of Transposition Ciphers Using Compression." Pp. 36–52 in International Conference on Cryptology and Network Security.

[5] Alabbas, Maytham and Allan Ramsay. 2012. "Improved POS-Tagging for Arabic by Combining Diverse Taggers." Pp. 107–16 in IFIP International Conference on Artificial Intelligence Applications and Innovations.

[6] Alghamdi, Mansoor A., Ibrahim S. Alkhazi, and William J. Teahan. 2016. "Arabic OCR Evaluation Tool." Pp. 1–6 in Computer Science and Information Technology (CSIT), 2016 7th International Conference on. IEEE.

[7] Alhawiti, Khaled M. 2014. "Adaptive Models of Arabic Text." Ph.D. thesis, Bangor University.

[8] Alkahtani, Saad. 2015. “Building and Verifying Parallel Corpora between Arabic and English.” Ph.D. thesis, Bangor University.

[9] Alkahtani, Saad and William J. Teahan. 2016. “A New Parallel Corpus of Arabic/English.” Pp. 279–84 in Proceedings of the Eighth Saudi Students Conference in the UK.

[10] Alkhazi, Ibrahim S., Mansoor A. Alghamdi, and William J. Teahan. 2017. “Tag Based Models for Arabic Text Compression.” Pp. 697–705 in 2017 Intelligent Systems Conference (IntelliSys). IEEE.

[11] Alkhazi, Ibrahim S. and William J. Teahan. 2017. “Classifying and Segmenting Classical and Modern Standard Arabic Using Minimum Cross-Entropy.” INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 8(4):421–30.

[12] Alkhazi, Ibrahim S. and William J. Teahan. 2018. “BAAC: Bangor Arabic Annotated Corpus.” INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 9(11):131–40.

[13] Alosaimy, Abdulrahman Mohammed S. 2018. “Ensemble Morphosyntactic Analyser for Classical Arabic.” Ph.D. thesis, University of Leeds.

[14] Alqrainy, Shihadeh. 2008. “A Morphological-Syntactical Analysis Approach for Arabic Textual Tagging.”

[15] Anon. n.d. “Madamira Arabic Analyzer - Online.” Retrieved February 17, 2019a (https://camel.abudhabi.nyu.edu/madamira/).

[16] Anon. n.d. “The Stanford Natural Language Processing Group.” Retrieved February 17, 2019b (https://nlp.stanford.edu/software/tagger.shtml).

[17] Atwell, Eric Steven, Salim Elsheikh, and Mohammad Elsheikh. 2018. “TIMELINE OF THE DEVELOPMENT OF ARABIC POS TAGGERS AND MORPHOLOGICALANALYSERS.”

[18] Brill, Eric. 1992. “A Simple Rule-Based Part of Speech Tagger.” Pp. 152–55 in Proceedings of the third conference on Applied natural language processing.

[19] Brown, Peter F., Vincent J. Della Pietra, Robert L. Mercer, Stephen A. Della Pietra, and Jennifer C. Lai. 1992. “An Estimate of an Upper Bound for the Entropy of English.” Computational Linguistics 18(1):31–40.

[20] Cleary, John and Witten, Ian. 1984. “Data Compression Using Adaptive Coding and Partial String Matching.” C(4):396–402.

[21] Columbia University. n.d. “Arabic Language Disambiguation for Natural Language Processing Applications - Cu14012 - Columbia Technology Ventures.” Retrieved (http://innovation.columbia.edu/technologies/cu14012_arabic-language-disambiguation-for-natural-language-processing-applications).

[22] Darwish, Kareem, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. “Multi-Dialect Arabic POS Tagging: A CRF Approach.” in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).

[23] Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. 2004. “Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks.” Pp. 149–52 in Proceedings of HLT-NAACL 2004: Short papers.

[24] Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. 2007. “Automatic Processing of Modern

Standard Arabic Text." Pp. 159–79 in Arabic Computational Morphology. Springer.

[25] Diab, Mona T. 2007. "Improved Arabic Base Phrase Chunking with a New Enriched POS Tag Set." Pp. 89–96 in Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources.

[26] El-Kareh, Seham and Sameh Al-Ansary. 2000. "An Interactive Multi-Features POS Tagger." P. 83Y88 in the Proceedings of the International Conference on Artificial and Computational Intelligence for Decision Control and Automation in Intelligence for Decision Control and Automation in Engineering and Industrial Applications.

[27] Francis, W. Nelson and Henry Kucera. 1979. "The Brown Corpus: A Standard Corpus of Present-Day Edited American English." Providence, RI: Department of Linguistics, Brown University [Producer and Distributor].

[28] Green, Spence and Cd Manning. 2010. "Better Arabic Parsing: Baselines, Evaluations, and Analysis." COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics (August):394–402.

[29] Green, Spence, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. "Parsing Models for Identifying Multiword Expressions." Computational Linguistics 39(1):195–227.

[30] Greene, Barbara B. and Gerald M. Rubin. 1971. "Automated Grammatical Tagging of English."

[31] Habash, Nizar and Owen Rambow. 2005. "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop." Pp. 573–80 in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.

[32] Habash, Nizar, Owen Rambow, and Ryan Roth. 2009. "MADA+ TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization." Pp. 102–9 in Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt.

[33] Habash, Nizar, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. "Morphological Analysis and Disambiguation for Dialectal Arabic." Pp. 426–32 in Hlt-Naacl.

[34] El Hadj, Yahya, I. Al-Sughayeir, and A. Al-Ansari. 2009. "Arabic Part-of-Speech Tagging Using the Sentence Structure." in Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt.

[35] Hadni, Meryeme, Said Alaoui Ouatik, Abdelmonaime Lachkar, and Mohammed Meknassi. 2013. "Hybrid Part-of-Speech Tagger for Non-Vocalized Arabic Text." International Journal on Natural Language Computing (IJNLC) Vol 2.

[36] Hajic, Jan, Otakar Smrz, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. "Prague Arabic Dependency Treebank: Development in Data and Tools." Pp. 110–17 in Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools.

[37] Jelinek, Fred. 1990. "Self-Organized Language Modeling for Speech Recognition." Readings in Speech Recognition 450–506.

[38] Katz, Slava. 1987. "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer." IEEE Transactions on Acoustics, Speech, and Signal Processing 35(3):400–401.

[39] Khmelev, Dmitry V and William J. Teahan. 2003. "A Repetition Based Measure for Verification of Text Collections and for Text Categorization." Pp. 104–10 in Proceedings of

the 26th annual international ACM SIGIR conference on Research and development in information retrieval. ACM.

[40] Khoja, Shereen. 2001. “APT: Arabic Part-of-Speech Tagger.” Pp. 20–25 in Proceedings of the Student Workshop at NAACL.

[41] Khoja, Shereen. 2003. “APT: An Automatic Arabic Part-of-Speech Tagger.” Ph.D. thesis, Lancaster University.

[42] Khoja, Shereen, Roger Garside, and Gerry Knowles. 2001. “A Tagset for the Morphosyntactic Tagging of Arabic.” Proceedings of the Corpus Linguistics. Lancaster University (UK) 13.

[43] Klein, Sheldon and Robert F. Simmons. 1963. “A Computational Approach to Grammatical Coding of English Words.” Journal of the ACM (JACM) 10(3):334–47.

[44] Kuhn, Roland and Renato De Mori. 1990. “A Cache-Based Natural Language Model for Speech Recognition.” IEEE Transactions on Pattern Analysis and Machine Intelligence 12(6):570–83.

[45] Linguistic Data Consortium. 2002. Buckwalter Arabic Morphological Analyzer : Version 1.0. Linguistic Data Consortium.

[46] Maamouri, Mohamed and Ann Bies. 2004. “Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools.” Pp. 2–9 in Proceedings of the Workshop on Computational Approaches to Arabic Script-based languages.

[47] Martinez, Angel R. 2012. “Part-of-Speech Tagging.” Wiley Interdisciplinary Reviews: Computational Statistics 4(1):107–13.

[48] Mohamed, Emad and Sandra Kübler. 2010. “Arabic Part of Speech Tagging.” in LREC.

[49] Nguyen, Dat Quoc, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham. 2014. “RDRPOSTagger: A Ripple down Rules-Based Part-of-Speech Tagger.” Pp. 17–20 in Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics.

[50] nltk.org. n.d. “Simple Pipeline Architecture for an Information Extraction System.” Retrieved February 8, 2019 (http://www.nltk.org/book/ch07.html).

[51] Pasha, Arfath, Mohamed Al-badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M. Roth. 2014. “MADAMIRA : A Fast , Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic.” Proceedings of the 9th Language Resources and Evaluation Conference (LREC’14) 14:1094–1101.

[52] Richards, Debbie. 2009. “Two Decades of Ripple down Rules Research.” The Knowledge Engineering Review 24(2):159–84.

[53] Al Shamsi, Fatma and Ahmed Guessoum. 2006. “A Hidden Markov Model-Based POS Tagger for Arabic.” Pp. 31–42 in Proceeding of the 8th International Conference on the Statistical Analysis of Textual Data, France.

[54] Soudi, Abdelhadi, Ali Farghaly, Günter Neumann, and Rabih Zbib. 2012. Challenges for Arabic Machine Translation. Vol. 9. John Benjamins Publishing.

[55] Taylor, Ann, Mitchell Marcus, and Beatrice Santorini. 2003. “The Penn Treebank: An Overview.” Pp. 5–22 in Treebanks. Springer.

[56] Teahan, W. J. and John G. Cleary. 1998. “Tag Based Models of English Text.” Pp. 43–52 in Data Compression Conference. IEEE.

[57] Teahan, William. 2018. “A Compression-Based Toolkit for Modelling and Processing Natural Language Text.” Information 9(12):294.

[58] Teahan, William J. and John G. Cleary. 1997. “Applying Compression to Natural Language Processing.” in SPAE: The Corpus of Spoken Professional American-English.

[59] Teahan, William J., Yingying Wen, Rodger McNab, and Ian H. Witten. 2000. “A Compression-Based Algorithm for Chinese Word Segmentation.” Computational Linguistics 26(3):375–93.

[60] Teahan, William John. 1998. “Modelling English Text.” Ph.D. thesis, Waikato University.

[61] Teahan, William John. 2000. “Text Classification and Segmentation Using Minimum Cross-Entropy.” Pp. 943–61 in Content-Based Multimedia Information Access-Volume 2.

[62] Teahan, William John, Stuart Inglis, John G. Cleary, and Geoffrey Holmes. 1998. “Correcting English Text Using PPM Models.” Pp. 289–98 in Data Compression Conference, 1998. DCC’98. Proceedings.

[63] Tim Buckwalter. n.d. “Buckwalter Arabic Transliteration.” Retrieved January 29, 2019 (http://www.qamus.org/transliteration.htm).

[64] Wintner, Shuly. 2014. “Morphological Processing of Semitic Languages.” Pp. 43–66 in Natural language processing of Semitic languages. Springer.

[65] Wu, Peiliang. 2007. “Adaptive Models of Chinese Text.” University of Wales, Bangor.