NUWT: JAWI-SPECIFIC BUCKWALTER CORPUS FOR MALAY WORD TOKENIZATION

[1]Juhaida Abu Bakar, [2]Khairuddin Omar , [2]Mohammad Faidzul Nasrudin & [2]Mohd Zamri Murah
[1] Universiti Utara Malaysia, Malaysia
[2] Universiti Kebangsaan Malaysia, Malaysia

juhaida.ab@uum.edu.my;ko@ukm.edu.my;mfn@ukm.edu.my;
zamri@ukm.edu.my

ABSTRACT

This paper describes the design and creation of a monolingual parallel corpus for the Malay language written in Jawi. This paper proposes a new corpus called the National University of Malaysia Word Tokenization (NUWT) corpora To the best of our knowledge, currently, there is no sufficiently comprehensive, well-designed standard corpus that is annotated and made available for the public for the Jawi script corpora. This corpus contains the Jawi-specific Buckwalter character code and can be used to evaluate the performance of word tokenization tasks, as well as further language processing. The objective of this work is to conform and standardize the corpora between similar characters in Jawi. It consists of three subcorporas with documents from different genres. The gathering and processing steps, as well as the definition of several evaluation tasks regarding the use of these corpora, are included in this paper. One of the important roles and fundamental tasks of the corpus, which is the tokenization, is also presented in this paper. The development of the Malay language tokenizer is based on the syntactic data compatibility of Malay words written in Jawi. A series of experiments were performed to validate the corpus and to fulfill the requirement of the Jawi script tokenizer with an average error rate of 0.020255. Based on this promising result, the token will be used for the disambiguation and unknown word resolution, such as out-of-vocabulary (OOV) problem in the tagging process.

Keywords: Malay corpora, word tokenization, regular expression, Buckwalter character code.

INTRODUCTION

The annotated corpora, the *Malay corpus* (Knowles & Don, 2003) and the *Dewan Bahasa dan Pustaka* (DBP) database (DBP, 2015) corpora, are among the most valuable resources in current natural language processing in Malay studies. They underlie statistical research in monolingual tasks, such as sentence structure, syntactic disambiguation, semantic recognition, information retrieval, etc. Annotated corpora constitutes a very useful tool for research.

The Malay language consists of two writing systems. The Malay language is usually written in Roman (Rumi), which stands for the Latin alphabet, and also written in Jawi, which is originally from the Arabic language. Some efforts are currently being undertaken to strengthen Jawi writing among the Malays in Malaysia. For the Malaysian Malay community, the creation of a monolingual parallel corpus between Roman and Jawi has a special significance. It provides the basis for the development of Malay language applications that can be used to facilitate or even avoid labor and time-consuming processes of manual handling of parallel language information. In addition, such a corpus enables the empowerment of minority languages. With the use of a monolingual parallel corpus and the methods which allow for the transfer of linguistic annotations across parallel languages, new resources and tools can be created for the minority languages.

The goal of the research as presented in this paper is for the development of a parallel language corpus and basic tools and resources for the Malay language. This paper describes the creation of such parallel corpus and the attachment of a part-of-speech (POS) tagset for the Malay corpus. The pattern of the Malay language is quite different from Indo-European languages because of its lower level. At high syntactic level, the language is similar enough to Indo-European language, and one can talk of direct objects in transitive constructions and even of agentless passive. The dominant sentence order is SVO (Knowles & Don, 2003).

Several collections of unannotated Jawi texts do exist. However, the only corpus with incorporated linguistic information that is currently available for the Malay language is a small corpus of approximately 30,500 tokens annotated with POS analyses. The Malay corpus written in Rumi are the *Malay corpus* (Mohamed, Omar, & Ab Aziz, 2011) and the *Malay Corpus UKM-DBP* (Saad, Bakar, Karim, Tukiman, & Nor, 2012). For the attachment of the POS tagset to a new parallel corpus, an original tagset applied in Mohamed et al. (2011)

and Saad et al. (2012) is used. To evaluate the new corpus, a pilot test was conducted on word tokenization, which is the first step of any kind of natural language text preparation.

LITERATURE REVIEW

In this section, an overview is given on related corpora, such as the English, Arabic and Austronesian corpus. The Natural Language ToolKit (NLTK) provides convenient ways to access several of the English corpora, with more than 38 corpora listed in the NLTK Data (NLTK Project, 2015). The NLTK provides many text corpora which contain linguistic annotations representing prosodic, part-of-speech tags, named entities, syntactic structures, semantic roles, etc. Table 1 sums up related corpora by giving a comparison between all these corpora.

Table 1

Comparison between Corpus

Corpus	Corpus Type	Language	Content	Availability
Brown Corpus	Grammatical corpus	English	1.15 million words, tagged and categorized 15 genres	Public
Penn Treebank	Grammatical corpus	English	40,000 words, tagged and parsed	Public (selection)
CMU Pronouncing Dictionary	Prosodic corpus	English	100,000 words and transcriptions	Public
WordNet 3.0	WordNet	English	145,000 synonym sets	Public
Quranic Arabic Corpus (Dukes & Habash, 2010)	Syntactic and morphological Quran corpus	Arabic (Arabic Buckwalter code)	Builds on the verified Arabic text distributed by the Tanzil project	Public
SEAlang Library Malay Text Corpus ("SEAlang Projects," 2011)	Monolingual corpus	Malay (Roman)	Consists of Malay texts retrieved from a variety of Internet sources	Proprietary
WordNet Bahasa	WordNet	Malay (Roman)	Malay semantic dictionary (Malaysian and Indonesian)	Proprietary
Quranic Malay written in Jawi character Corpus (Sulaiman, 2013)	Monolingual unannotated corpus	Malay (Jawi)	157,388 words	Upon request

(continued)

Corpus	Corpus Type	Language	Content	Availability
Malay corpus (Mohamed et al., 2011)	Grammatical corpus	Malay (Roman)	18,135 tokens	Upon request
Malay corpus UKM-DBP (Saad et al., 2012)	Grammatical corpus	Malay (Roman)	12,304 words	Upon request
NUWT Corpus	Grammatical literary corpus	Malay (Jawi-specific Buckwalter)	187,827 words	Commercial market

Unannotated Jawi corpus has been developed from a variety of sources, including old Jawi manuscripts, Al-Quran text translation, textbooks and local newspapers. Research on Jawi is widely used in the learning field, for instance in Multimedia (MM), computer hardware, such as virtual keyboard (Engineering) and in the field of Artificial Intelligence in the study of Pattern Recognition (PR), Natural Language Processing (NLP) and Machine Learning (ML). In the study of PR, a Jawi script undergoes the process of character recognition in handwritten or printed form. Segmentation deals with the identification of the Jawi characters. Then, the characters are dealt with in the processes of NLP and ML field at the final part. Previous studies related to Jawi include the research on MM (Diah, Ismail, Ahmad, & Abdullah, 2010; Diah, Ismail, Hami, & Ahmad, 2011); Engineering (Ismail, Yusof, & Jomhari, 2010); PR (Azmi, 2013; Heryanto, Nasrudin, & Omar, 2008; Redika, Omar, & Nasrudin, 2008); and NLP (C. W. S. B. C. W. Ahmad, Omar, Nasrudin, Murah, & Azmi, 2013; Sulaiman, 2013), just to name a few. Figure 1 shows the trend of the studies conducted on the Jawi script.

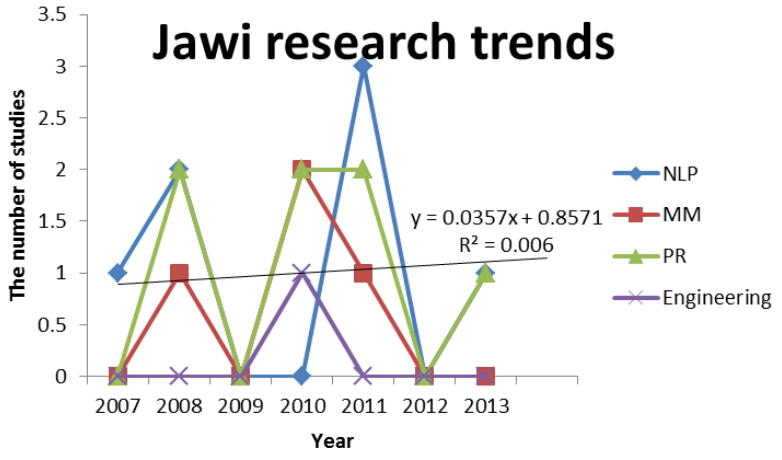

Figure 1. Jawi research trends and relationship to the field.

The corpus development of the Jawi script starts from the use of the POS tagset. For the Malay language written in Roman, a number of researches pioneered by linguists and computing researchers have developed a POS tagset corpus in justification of their research. Atwell (2008) states that before we develop a POS tagset, or decide to re-use an existing pos-tagset, we should be clear about why we want to pos-tag the corpus. For developers of corpus resources for general purposes, the aim is perhaps to enrich the text with linguistic analyses to maximize the potential for corpus re-use in a wide range of applications. On the other hand, very fine-grained distinctions may cause problems for automatic tagging if some words can change the grammatical tag depending on the function and context. At present, POS tagset for Jawi is no longer being developed. In order to do that, we believe that the corpus development of the Jawi script can facilitate the task of analyzing the NLP task pipeline for the Malay language written in Jawi. In the next section, the detailed description of the new corpus, namely the NUWT corpora, its corresponding collection, its acquisition processes as well as its structure, is given.

NUWT CORPORA DESCRIPTION

The NUWT corpora sources were gathered from three different genres of documents. The "Quranic Malay written in Jawi character Corpus" (Sulaiman, Omar, Omar, Murah, & Rahman, 2011) is an unannotated text of the Quran translation and contains a collection of 114 chapters with 157,388 words. The corpus was used on the NLP task, Stemmer for Jawi characters, using two sets of rules in Jawi. One set of rules was used to stem various forms of derived words, while the other set was used to replace the use of a dictionary by producing the root word for each derivative.

The second source is an annotated corpus named the "Malay corpus" and contains 18,135 tokens with 1,381 words that have ambiguous tags. The corpora was prepared by Mohamed et al. (2011) using the Dewan Bahasa dan Pustaka (DBP) tagset. The DBP tagset was used because it is the highest government authorized body concerning Bahasa Malaysia and the grammar follows that of Bahasa Malaysia. The "Malay corpus" was written using Roman (Rumi) writing and has 21 tags, as shown in Table 2.

The third source corpus is a grammatical corpus named the "Malay corpus UKM-DBP". It is retrieved from Saad et al. (2012) and is a collection of newspapers, magazines and books with 12,304 words. The corpus was developed according to the DBP tagset and written using Roman writing. It has five main tags, with the elaboration fraction for each main tag shown in Table 3.

Table 2

Malay Tagset DBP in Malay Corpus

Tagset	Description in Malay	Description in English	Example
KN	*Kata Nama*	Noun	meja/*table*, kerusi/*chair*
KK	*Kata Kerja*	Verb	makan/*eat*, tidur/*sleep*
ADJ	*Kata Adjektif*	Adjective	hitam/*black*, cantik/*beautiful*, dalam/*deep*
KSN	*Kata Sendi Nama*	Preposition	di/*at/on/in*, ke/*to,*dari/*from*, kepada/*to/towards/at*, dalam/*in*
KB	*Kata Bantu*	Auxiliary verb	akan/*will/shall*, belum/*not yet,* boleh/*can/may*, telah/*already*
KG	*Kata Ganti Nama*	Pronoun	saya/*I/me*, awak/*you*
KH	*Kata Hubung*	Conjunction	yang/*null*, dan/*and*, atau/*or*
ADV	*Kata Adverba*	Adverb	bahawasanya/*in fact/truly*, barangkali/*may be/probably*
KT	*Kata Tanya*	Question	apa/*what*, berapa/*how much/how many*, mengapa/*why*
KBIL	*Kata Bilangan*	Cardinal	satu/*one*, dua/*two*
KPM	*Kata Pemeri*	Narrator	adalah/*is/are*, ialah/*is/are*
KP	*Kata Perintah*	Command	jangan/*don't*, sila/*please*
KAR	*Kata Arah*	Direction	dalam/*inside*, atas/*on top*, bawah/*under*
PW	*Penanda Wacana*	Discourse mark	hatta/*then*, maka/*thus*
KEP	*Kata Kependekan*	Short form	UNCR, PBB
#E	*Klitik –lah*	Clitic *lah*	cubalah/*please try*
@KG	*Klitik –nya*	Clitic *nya*	bukunya/*his book*
KNF	*Kata Nafi*	Deny	tidak/*no*, bukan/*not*
KNK	*Kata Nama Khas*	Proper noun	Abdullah Badawi
SEN	*Senarai nombor*	List number	(i), (ii), (iii), etc
SYM	*Simbol atau tanda baca*	Any symbol or punctuations	. , " - + etc

Source: *Mohamed et al. (2011)*

Table 3.

Malay DBP Tagset in the Malay UKM-DBP Corpus

Tagset	Description in Malay	Description in English	Detailed Description in Malay	Detailed Description in English
NAK	*Kata Nama*	Noun	*Kata Nama Am Konkrit*	Concrete Noun
NAA			*Kata Nama Am Abstrak*	Abstract Noun
NKK			*Kata Nama Khas Konkrit*	Concrete Proper Noun
NKA			*Kata Nama Khas Abstrak*	Abstract Proper Noun
VT	*Kata Kerja*	Verb	*Kata Kerja Transitif*	Transitive Verb
VTT			*Kata Kerja Tak Transitif*	Intransitive Verb
VB			*Kata Kerja Bantu*	Auxiliary verb
A	*Kata Adjektif*	Adjective	*Kata Adjektif*	Adjective
PR	*Kata Preposisi*	Preposition	*Kata Preposisi*	Preposition
G1	*Kata Ganti Nama*	Pronoun	*Kata Ganti Nama Pertama*	Pronoun #1
G2			*Kata Ganti Nama Kedua*	Pronoun #2
G3			*Kata Ganti Nama Ketiga*	Pronoun #3

NUWT CORPORA ANALYSIS

Orthography is the main problem that has led to the development of the NUWT Corpora. Many NLP researchers have focused on Roman script (Abdullah, Hashim, & Mohamed Husin, 2011; Mohamed et al., 2011; Noor, Noah, Aziz, & Hamzah, 2010; Suhaimi Ab Rahman, Omar, & Aziz, 2011; Suhaimi Ab Rahman, Omar, Mohamed, Juzaidin, & Aziz, 2011; Suhaimi Abdul Rahman & Omar, 2013); while just a few have focused on Jawi script (Abdul Ghani, Zakaria, & Omar, 2009; Abu Bakar, 2008; C. W. S. B. C. W. Ahmad et al., 2013; C. W. S. C. W. Ahmad, 2007; Sulaiman, Omar, Omar, Murah, & Abdul Rahman, 2014; Yonhendri, 2008). Figure 2 shows the original Jawi orthography from old manuscripts.

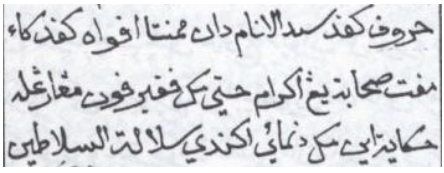

Figure 2. Original Jawi orthography

Close orthography, such as the Arabic language, leads in such studies (Diab, Hacioglu, & Jurafsky, 2004). Even though Jawi is similar to Arabic, Jawi has six more characters than Arabic. Jawi script also represents the Malay language, which is different from Malay (Roman). MADAMIRA is an Arabic morphology project by Pasha et al. (2014), which can be accessed at *http://nlp.ldeo.columbia.edu/madamira/*. Figure 3 shows a sample input and Figure 4 shows the result of the tokenization and morphology from MADAMIRA. As shown, several additional characters have been changed to the equivalent similar orthography in Arabic.

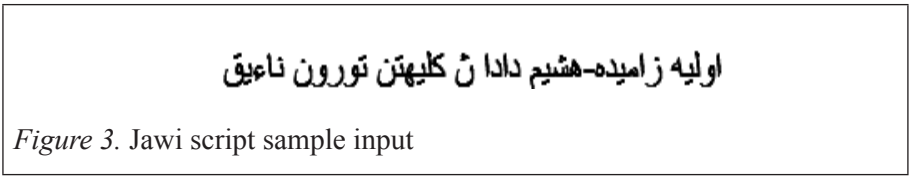

Figure 3. Jawi script sample input

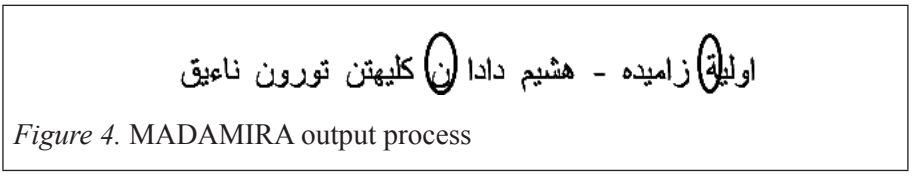

Figure 4. MADAMIRA output process

However, the equivalent similar orthography represents another pronunciation sound of the word. The orthography issues are shown in Table 4.

Table 4.

Orthographic issues using MADAMIRA

Malay Orthography	Roman equivalent sound	Arabic Orthography	Roman equivalent sound
ه	h	ة	t/h
ڽ	ny	ن	n
ۏ	v	و	w, u, o
ڠ	ng	غ	gh
ݢ	g	ک	k
ڤ	p	ف	f

NUWT CORPORA DEVELOPMENT

The NUWT corpora development goes through several phases, such as pre-processing or encoding phase, tokenization and corpus annotation. In the first phase, two encoding transliteration processes are carried out, which are the

transliteration of Roman to Jawi and the transliteration of Jawi (Unicode) to Jawi (Buckwalter code). In this phase, both the Malay corpora (*Malay corpus and Malay corpus UKM-DBP)* are translated into Jawi using *Teruja* or *Ejawi* (*Malay Text Rumi-To-Jawi Script Transliteration system)*, which can be accessed at *http://www.jawi.ukm.my* and *http://www.ejawi.net/*. After the transliteration process, the data is randomly checked using the Rumi-Jawi-Unicode dictionary for writing style and Unicode. Table 5 shows several errors detected using *Teruja* and *Ejawi*. The processed data is shown in Fig. 5.

Table 5

Errors after using the Online Transliteration System

Error	Description
Letters	Wrong letter used between Heh ه and Hah ح, Theh ة and Teh ت, Kaf ک and Qaf ق, Yeh ي and Alef Maksura ى, and Alef ا with Alef, Hamza Above أ or Alef, Hamza Below إ
Loanword	The suggested word is given but requires for a human expert to check the spelling.
Variant positional forms	An Arabic representation form has variant positional forms, which contains isolated, medial, initial and final forms. All Arabic texts (including Jawi) follow Arabic representation form. In order to use Jawi→Buckwalter transliteration code, we only used Arabic letter in the Arabic block (U+0600..U+06FF) or the Arabic Supplement block (U+0750..U+077F).

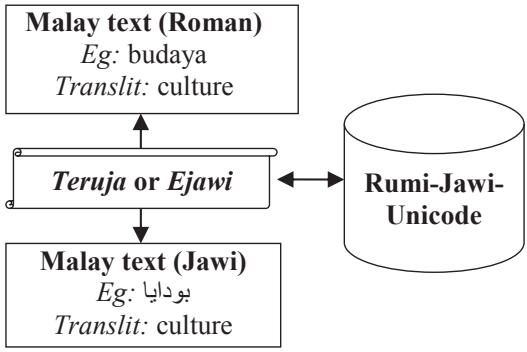

Figure 5. Pre-processing phase in preparing the data.

In the second phase, the NUWT Corpora scripts are translated into Buckwalter transliteration. The Roman (Rumi) writing uses ASCII character code. Meanwhile, Jawi writing uses UNICODE character code[1]. Jawi script has 36 characters, consisting of 30 shared characters with the Arabic language and six newly created characters to meet the needs of Malay language phonemes. The six additional letters are ca چ, nga ڠ, nya ڽ, pa ڤ, ga ڬ, and va ۏ. In this project, we used the Buckwalter transliteration to the Jawi corpora[2], named NUWT Corpora.

The Buckwalter transliteration has been used in many publications in natural language processing (NLP) and in resources developed at the LDC (Habash, Soudi, & Buckwalter, 2007). The main advantages of the Buckwalter transliteration are that it is a strict transliteration (i.e., one-to-one) and it is written in ASCII characters. Similar orthographic languages, such as Urdu (Habash & Metsky, 2008) and Pakistani language (Irvine, Weese, & Callison-Burch, 2012) also use Buckwalter transliteration. The extended Buckwalter transliteration applied to Jawi scripts is shown in Table 6 (Abdul Rahman, 1999; Unicode, 2014). The highlighted sections indicate those parts of the scheme that have been extended over the original scheme.

Table 6

The Positional Variant Forms of Jawi Characters with Unicode & Buckwalter

Letter	Unicode (U+)	Description (Letter)	Roman equivalent	Isolated Form	Unicode (U+)	Initial Form	Unicode (U+)	Medial Form	Unicode (U+)	Final Form	Unicode (U+)	Buckwalter
ا	0627	Alef	a	ا	FE8D					ـا	FE8E	A
ب	0628	Beh	b	ب	FE8F	بـ	FE91	ـبـ	FE92	ـب	FE90	b
ة	0629	Teh Marbuta	t/h	ة	FE93					ـة	FE94	p
ت	062A	Teh	t	ت	FE95	تـ	FE97	ـتـ	FE98	ـت	FE96	t
ث	062B	Theh	s, (th)	ث	FE99	ثـ	FE9B	ـثـ	FE9C	ـث	FE9A	v
ج	062C	Jeem	j	ج	FE9D	جـ	FE9F	ـجـ	FEA0	ـج	FE9E	j
ح	062D	Hah	h, (h∞)	ح	FEA1	حـ	FEA3	ـحـ	FEA4	ـح	FEA2	H
چ	0686	Tcheh	c	چ	FB7A	چـ	FB7C	ـچـ	FB7D	ـچ	FB7B	J
خ	062E	Khah	kh	خ	FEA5	خـ	FEA7	ـخـ	FEA8	ـخ	FEA6	x
د	062F	Dal	d	د	FEA9					ـد	FEAA	d
ذ	0630	Thal	z, (dh)	ذ	FEAB					ـذ	FEAC	*

(continued)

1 http://unicode.org/charts/PDF/Unicode-7.0 (Range: 0600-06FF, FB50-FDFF, FE70-FEFF)

2 All Arabic script transliterations are provided in the Extended Buckwalter transliteration scheme (Bakar, Omar, Nasrudin, Murah, & Ahmad, 2013). This scheme extends Buckwalter's transliteration scheme (Buckwalter, 2002) to increase its readability while maintaining the one-to-one correspondence with the orthography as represented in Unicode. For Jawi-specific extensions of the Arabic scripts, we extend the (Habash et al., 2007) transliteration scheme as follows:
X ڽ, W ۏ, e ڠ, Q ک

Letter	Unicode (U+)	Description (Letter)	Roman equivalent	Isolated Form	Unicode (U+)	Initial Form	Unicode (U+)	Medial Form	Unicode (U+)	Final Form	Unicode (U+)	Buckwalter
ر	0631	Reh	r	ر	FEAD					ر	FEAE	r
ز	0632	Zain	z	ز	FEAF					ز	FEB0	z
س	0633	Seen	s	س	FEB1	سـ	FEB3	ـسـ	FEB4	ـس	FEB2	s
ش	0634	Sheen	sy, (sh)	ش	FEB5	شـ	FEB7	ـشـ	FEB8	ـش	FEB6	$
ص	0635	Sad	s, (s∞)	ص	FEB9	صـ	FEBB	ـصـ	FEBC	ـص	FEBA	S
ض	0636	Dad	d, (d∞)	ض	FEBD	ضـ	FEBF	ـضـ	FEC0	ـض	FEBE	D
ط	0637	Tah	t, (t∞)	ط	FEC1	طـ	FEC3	ـطـ	FEC4	ـط	FEC2	T
ظ	0638	Zah	z, (z∞)	ظ	FEC5	ظـ	FEC7	ـظـ	FEC8	ـظ	FEC6	Z
ع	0639	Ain	Initial: a,i,u Final: k, (‘)	ع	FEC9	عـ	FECB	ـعـ	FECC	ـع	FECA	E
غ	063A	Ghain	gh	غ	FECD	غـ	FECF	ـغـ	FED0	ـغ	FECE	g
ڠ	06A0	Ain with three dots above	ng									e
ف	0641	Feh	f	ف	FED1	فـ	FED3	ـفـ	FED4	ـف	FED2	f
ڤ	06A4	Veh	p	ڤ	FB6A	ڤـ	FB6C	ـڤـ	FB6D	ـڤ	FB6B	V
ق	0642	Qaf	k, q, (q)	ق	FED5	قـ	FED7	ـقـ	FED8	ـق	FED6	q
ک	06A9	Kaf	k	ك	FED9	كـ	FEDB	ـكـ	FEDC	ـك	FEDA	k
ڬ	0762	Kaf with dot above	g									Q
ل	0644	Lam	l	ل	FEDD	لـ	FEEF	ـلـ	FEE0	ـل	FEDE	l
م	0645	Meem	m	م	FEE1	مـ	FEE3	ـمـ	FEE4	ـم	FEE2	m
ن	0646	Noon	n	ن	FEE5	نـ	FEE7	ـنـ	FEE8	ـن	FEE6	n
ه	0647	Heh	h	ه	FEE9	هـ	FEEB	ـهـ	FEEC	ـه	FEEA	h
و	0648	Waw	w, u, o	و	FEED					ـو	FEEE	w
ۏ	06CF	Waw with Dot Above	v									W
ي	064A	Yeh	y, i, e taling	ي	FEF1	يـ	FEF3	ـيـ	FEF4	ـي	FEF2	y
ى	0649	Alef Maksura	final e pepet	ى	FEEF					ـى	FEF0	Y
ڽ	06BD	Noon with three dots above	ny									X
ء	0621	Hamza	Initial: drop Final: k, (‘)	ء	FE80							’
أ	0623	Alef, Hamza above	-	أ	FE83					ـأ	FE84	>
إ	0625	Alef, Hamza Below	-	إ	FE87					ـإ	FE88	<
ئ	0626	Yeh, Hamza above	-	ئ	FE89	ئـ	FE8B	ـئـ	FE8C	ـئ	FE8A	}
٢	0662	Indic Digit Two	-									2

The next phase in natural language text preparation is tokenization, which typically plays an important role in cutting a string into identifiable units that constitute a piece of language data (Bird, Klein, & Loper, 2009). The simplest method commonly used in tokenizing a text is to split it on whitespace. Although this is the fundamental task in NLP, Jawi script is still far from having a standard tokenizer. At present, there is no sufficiently comprehensive, well-designed standard corpus that is annotated and publicly available for the

Jawi script corpora. Tokenization task is chosen to evaluate the significance of the corpus because of the Buckwalter transliteration format applied to Jawi characters. Further explanation is made on the NUWT Corpora Evaluation.

Corpus Annotation

POS annotation (also morpho-syntactic annotation) consists of assigning each word in a corpus to its general word-class (e.g., nouns), or to finer-grained grammatical categories (e.g., singular common noun) (Balossi, 2014). It also enables the tagging of homographs; for example, the term 'work' can be tagged as a verb or as a noun. Semantic annotation marks the semantic categories of words in a text; for example, the term 'bank' can belong to two different semantic fields according to whether it refers to a financial institution or an area of land along a river. Through lexical annotation, we learn about the lemma – the base form of a word – of each word form in our corpus; for example, 'speak', 'speaks', 'spoke, 'speaking' are forms of the same lexeme, with 'speak' as their lemma. Pragmatic annotation adds information to the words and multi-word expressions in a spoken conversation or dialogue; so the expression 'go now' may become a command or a question depending on the punctuation marks used. Linguistic annotation is used to capture a range of higher-level phenomena, including the practice of tagging the types of speech and thought presentation (e.g., direct speech and indirect speech, direct thought and indirect thought, etc.). On the use of corpus annotation, Leech (2005) states that "the practice of adding interpretative linguistic information to a corpus" confers an "added value" to it. An annotated corpus has the advantage of being used in either a "manual examination" or in an "automatic analysis". Moreover, it can be re-used and exploited for different aims and applications.

Both the *Malay corpus* and the *Malay corpus UKM-DBP* were developed according to the DBP tagset. General word-class has been used for both corpora. The Malay tagged corpus has been developed with the same tagset used in the modified bilingual dictionary (Hock, 2009) as stated in Mohamed et al. (2011). Some tags have not been used in the corpus, while a few new ones have been added. The AWL and KEP tags, which are linguistically not word classes, have not been used in the corpus. Other than that, clitics in Malay, such as *nya* (it, them), *mu* (you), *lah* (a particle added to words (suffix) used for emphasizing its predecessor word or sentence), *kah* (a particle at the end of a word or phrase for expressing enquiry), etc., are crucial to the Malay language. In the *Malay corpus*, only two clitics are handled, whereby the clitic, *nya* and *lah* are split into two tokens. For example, the word *terjejasnya* (it being affected) is split into *terjejas* (is affected) and *nya* (it). Thus, the tag @KG is used to tag *nya*, and #E for tagging *lah* (Mohamed et al., 2011). Other

added tags have also been used, such as KNF to tag denial words, KNK to tag proper nouns, SEN to tag any list numbers, and SYM to tag any symbols, including punctuations. In the NUWT Corpora, each tag used in the original corpus is maintained and the use of ASCII letter is also maintained as in the original corpus. Figures 6, 7 and 8 show the sample data of three corpus in the NUWT Corpus.

…1.1 den nAm Allh ye Vmwrh dAn VXAye
… (1.1 Dengan nama Allah yang Maha Pemurah dan Penyayang)
Gloss: In the name of Allah, the Beneficient, the Merciful

….1,2 sQnV Vwjyn Awntwq Allh , twhn smstA EAlm…
... (1,2 Segenap pujian untuk Allah, tuhan semesta Alam…)
Gloss: All praise is due to Allah, the Lord of the Worlds ...

Figure 6. The Quranic Malay written in the Jawi character corpus.

… nAmwn sAyA mndAVt …
KH KG KK
...(namun saya mendapat ...)
KH KG KK
Gloss: but I get ...

…dAVt mewrekn ms}lh2 sVrty…
KB KK KN KSN
... (dapat mengurangkan masalah-masalah seperti ...)
KB KK KN KSN
Gloss: may reduce problems such as ...

Figure 7. The Malay **c**orpus.

… rAVt s>wlh Vty-kArwn tq …
A NAK
... (rapat seolah peti-karun tq ...)
A NAK
Gloss: close like a treasure chest not ...

… EAlm AdAlh ktAb trbwk…
NAK NAK VTT
... (Alam adalah kitab terbuka ...)
NAK NAK VTT
Gloss: Nature is an open book ...

Figure 8. The Malay UKM-DBP corpus.

NUWT CORPORA EVALUATION

To evaluate the new corpus, previous researchers (Outahajala, Zenkouar, Benajiba, Rosso, & Elirf, 2013; Tmshkina, 2006) have used or applied the new corpus to the state-of-the-art POS tagger, such as TnT, Support Vector Machine (SVM) and Conditional Random Field (CRF). Accuracy (%) has been used to measure the performance of the corpus. However, in this study, a new orthography is used, which is the Jawi-specific Buckwalter transliteration, which is tagged along with the previous known tag set. Thus, it is believed that the tokenizer task is a suitable task for the evaluation of the corpora because of its orthographical differences with the Malay language written in Roman.

As a pilot study, the application of the NUWT corpus to a tokenization task was made to reveal what modifications are needed. The NLTK was used which is a platform for building Python programs to work with human language data. The NLTK is easy to use and interfaces with over 50 corpora and lexical resources, such as WordNet, and has a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing and semantic reasoning (NLTK Project, 2015). There are several types of tokenizers in NLTK, which are *Punkt Sentence Tokenizer, Regular Expression Tokenizer, S-Expression Tokenizer, Simple Tokenizer, Penn Treebank Tokenizer* and *NLTK Tokenizer package.* In our experiment, we used the regular expression tokenizer module from NLTK for the Jawi corpus. The *regular expression tokenizer* is deemed suitable for this task because of its uniqueness in handling language-dependent applications and algorithms (Shaalan & Raza, 2007; Sharum, Abdullah, Sulaiman, Murad, & Hamzah, 2011).

Jawi Regular-Expression Tokenizer (Jawi RegExpTokenizer)

A RegExpTokenizer splits substrings using a regular expression. A tokenizer then matches either the tokens or the separators between tokens. The tokenizer tokenizes a string, treating any sequence of blank lines as a delimiter. Blank lines are defined as lines containing no characters, except for space or tab characters (NLTK Project, 2015). The NLTK's regular expression tokenizer uses four parameters.

- pattern (str) - The pattern used to build this tokenizer. This pattern may safely contain capturing parentheses.
- gaps (bool) - True if this tokenizer's pattern should be used to find separators between tokens; False if this tokenizer's pattern should be used to find the tokens themselves.
- discard_empty (bool) - True if any empty tokens generated by the tokenizer should be discarded. Empty tokens can only be generated if _gaps == True.

- flags (int) - The regular expression flags used to compile this tokenizer's pattern. By default, the following flags are used: re.UNICODE | re.MULTILINE | re.DOTALL

Jawi-specific extension Buckwalter character code is normally not very different from English character code. By default, abbreviation patterns, e.g., TUDM (ت.م.د.وا *t.Aw.d.m*), MCA (مي.ع.ي.أ.يس *Eym.sy.>y*), have the same pattern or writing style as the English language, e.g., U.S.A. The abbreviations and acronyms in the Malay language are shown in Table 7 (Abdul Rahman, 1999; DBP, 2008). Arabic characters have no distinct uppercase and lowercase letter forms when writing abbreviations and acronyms.

Table 7

Writing Abbreviations and Acronyms for Jawi and Roman in Malay Language

No	Form	Roman	Jawi
1.	Special name; name of the position, rank, title, and proper name.	Profesor	روس.يف.ور.ڤ
2.	Initial abbreviation for the name of department and the position of the Malay language.	TUDM	ت.م.د.وا
3.	Initial abbreviation for the name of the department and the position of the English language.	MCA	مي.ع.ي.أ.يس
4.	Acronyms special name / common in Malay / English	Pas	س.اڤ

Patterns 1, 2, 3, 11, 12 and 13 (*non-highlighted table*) are common in the English language (refer to Table 8). Patterns 4-10 (*highlighted table*) are new patterns matching the Jawi corpus. If the pattern detects any punctuation marks, such as periods (.), commas (,), semicolons (;), colons (:), parentheses (), dashes (-), exclamation points (!), quotation marks (“) or underscores (_), the tokenizer will automatically segment the word into two separate words. Otherwise, the tokenizer will leave it as a single word. Patterns, such as * ذ, $ ش, > أ, < إ, and } ئ are supposed to be segmented as a single word when it occurs in the raw text or sentence. Patterns identified as a single or a separate word in Malay in the Jawi character are shown in Table 8. Regular expression patterns are sorted by priority. If a standard tokenizer built for other languages is used, such as the English language, patterns, such as *, >, <, }, will be segmented into two separate tokens. In the Arabic language, the orthographic problem interferes, as we have mentioned above.

Table 8

Tokenization Pattern for Jawi

No	Pattern	Segment taken
1.	Abbreviation, e.g., U.S.A	Single token
2.	Cardinal number, e.g., 1.1	Single token
3.	Words with internal hyphens, e.g., *Anq-Anq*, i.e., children	Single token
4.	Words with an asterisk (*) symbol (represent Thal (ذ))	Single token
5.	Words with a Dollar sign symbol $ (represent Syin (ش))	Single token
6.	Words with > sign symbol (represent letter alef with hamza above (أ))	Single token
7.	Words with < sign symbol (represent letter alef with hamza below (إ))	Single token
8.	Words with } sign symbol (represent letter yeh with hamza above (ئ))	Single token
9.	Words with Malaysian currency (RM)	Single token
10.	Single words other than the cases above	Single token
11.	Words with percentage (%)	Single token
12.	Ellipsis, e.g., …	Single token
13.	Any punctuation mark, such as .,;”?():-_!	Separate token

Algorithm and Implementation

The tokenizer process is summarized as per the following algorithm:

The following algorithm, *PatternMatching,* constructs tokens using regular expression

```
Require: text T, pattern P, gaps G,discard_empty D, flags F
begin
read sequence of T,
read regular expression pattern P,
  Pattern_Type1 : Abbreviation,
  Pattern_Type2 : Cardinal numbers,
  Pattern_Type3 : Hyphens
  Pattern_Type4 : Ellipsis
  Pattern_Type5 : Special character
    *$<>}’RM’
  Pattern_Type6 : Punctuation marks
read gaps G,read discard_empty D,
read flags F,
If found regular expression pattern P in sequence of T
  single token
    else
  separate token
applied gaps G, discard_empty D,
flags F
end
```

Tokenizer Experiments

The experiments outlined in this paper were tested using the bootstrapping approach and started with a few training words, 1,000, 5,000, 10,000 and 15,000 words for each corpus. A series of experiments were performed. Our first experiment was to manipulate a standard regular expression tokenizer built for the English language. One of the issues in tokenizing English words is the presence of contractions, such as *didn't*. In the NUWT Corpora, the apostrophe symbol (') represents the letter Hamza (ء). Other than that, the English language uses uppercase letters to identify abbreviations; meanwhile, in our corpora, the uppercase and lowercase letters for the abbreviations were used.

For the first implementation, the regular expression pattern suitable for Jawi-specific Buckwalter code was built up. The errors obtained from the first experiment were with a minimum error of 0.01038961, a maximum error of 0.029490617 and an average error of 0.020255. Several errors were obtained from a number of issues, as listed in Table 9.

Table 9

Errors from Certain Issues

Error	Action taken	Action that should have been taken
Words with asterisk (*) symbol, representing the letter Thal (ذ)	Missing word	Single word
Words with Dollar ($) sign, representing the letter Sheen (ش)	Separate word	Single word
Words with > sign symbol , representing the letter alef with hamza above (أ)	Missing word	Single word
Words with < sign symbol, representing the letter alef with hamza below (إ)	Missing word	Single word
Words with } sign symbol, representing the letter yeh with hamza above (ئ)	Missing word	Single word
Words with Malaysian currency (RM)	Separate word	Single word
Punctuation marks (.)	Single word	Separate word

For example, the * ذ (Thal) character can be seen in the front, middle, and final position of a word in Table 10. During the first experiment, only Thal in the middle position could be detected as a single word. However, Thal in the front and final part of the words is missing. See Figures 9, 10 and 11. Fig. 12 shows the error rate (%) for the NUWT corpus.

Table 10.

** ذ (Thal) in the Word*

Jawi	Transliteration	Gloss
تيروذ	*wryt	Offspring
ذاتسا	AstA*	Teacher
باذع	e*Ab	Penalty

Input: ttAVy kwAs Allh ye Ayeyn mmbry *wryp
Jawi: تتاڤي كواس الله يغ ايغين ممبري ذورية
Gloss: but the power of Allah to give offspring
Output: ttAVy | kwAs | Allh | ye | Ayeyn | mmbry | wryp

Figure 9. Error in program (First letter Thal ذ).

Input: kAlAw stAkt jAdy AstA* AtAw AmAm msjd
Jawi: كالاو ستاكت جادي استاذ اتاو امام مسجد
Gloss: if only become teacher or imam of the mosque
Output: kAlAw | stAkt | jAdy | AstA | AtAw | AmAm | msjd

Figure 10. Error in program (Last letter Thal ذ).

Input: dAn Awntwq mryk syqsA'n (E*Ab) ye Vdyh
Jawi: دان اونتوق مريک سيقساءن (عذاب) يغ ڤديه
Gloss: and there is a great punishment
Output: dAn | Awntwq | mryk | syqsA'n | (| E*Ab |) | ye | Vdyh

Figure 11. Middle letter Thal ذ detected the same as the input.

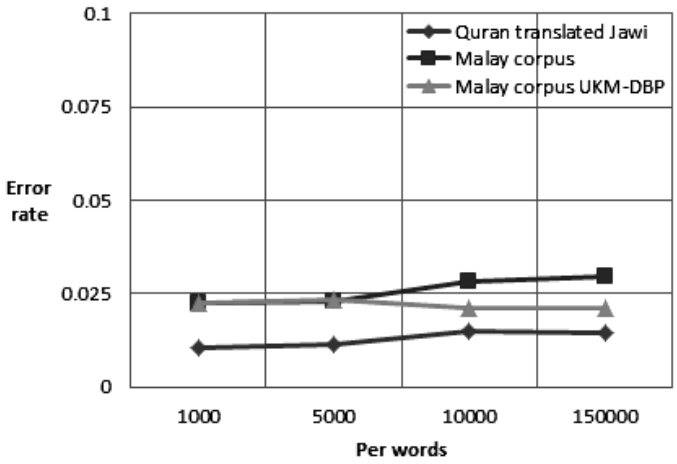

Figure 12. Error rate (%) for NUWT corpus.

After several modifications were made to the program's code, these errors were addressed. The second experiment shows the complete performance of the tokenization on the NUWT Corpora. The accuracy obtained was approximately 99.8%. For example, Figures 13 and 14 show the correct tokenizer word for Error 1. The output overcame the errors which occurred in the program for the same sentences (see Figures 9 and 10). The summary of successful tasks is shown in Table 11.

Input: ttAVy kwAs Allh ye Ayeyn mmbry *wryp
Jawi: تتافي كواس الله يغ ايغين ممبري ذورية
Gloss: but the power of Allah to give offspring
Output:

ttAVy	kwAs	Allh	ye	Ayeyn	mmbry	*wryp

Figure 13. First letter Thal ذ detected the same as the input.

Input: kAlAw stAkt jAdy AstA* AtAw AmAm msjd.
Jawi: كالاو ستاكت جادي استاذ اتاو امام مسجد
Gloss: if only become teacher or imam of the mosque
Output:

kAlAw	stAkt	jAdy	AstA*	AtAw	AmAm	msjd	.

Figure 14. Last letter Thal ذ detected the same as the input.

Table 11

Successful Tasks

Error	Action taken	Action that should have been taken
Words with asterisk (*) symbol, representing the letter Thal (ذ)	Single word	Single word
Words with Dollar ($) sign, representing the letter Sheen (ش)	Single word	Single word
Words with > sign symbol , representing the letter alef with hamza above (أ)	Single word	Single word
Words with < sign symbol, representing the letter alef with hamza below (إ)	Single word	Single word
Words with } sign symbol , representing the letter yeh with hamza above (ئ)	Single word	Single word
Words with Malaysian currency (RM)	Single word	Single word
Punctuation marks (.)	Separate word	Separate word

CONCLUSION AND FUTURE WORK

A corpora that contains three Malay subcorpora is described in this paper. This corpora is unique due to the code applied. This corpora will serve as a benchmarking corpus for the development and evaluation systems in word tokenization, as well as further language processing. A pilot study successfully developed a tokenizer system to reveal what modifications need to be made in the NUWT Corpora. We attempted to develop the tokenizer model for the NUWT Corpora using regular expression. The aim of this model is to suit the new corpora developed for the Malay language. Based on the experimental results and analysis, it can be concluded that regular expression is appropriate as a Jawi tokenizer in a Buckwalter transliteration format. Further works should focus on Out-Of-Vocabulary (OOV) problem in the POS tagging which is an important text analysis task that is used to classify words into their parts of speech. It labels them according to their tagsets, which is a collection of tags used for POS tagging.

ACKNOWLEDGMENTS

The first author is funded under the Skim Latihan Akademik IPTA-UUM (SLAI) by the Ministry of Higher Education Malaysia. We would like to thank Dr. Suliana Sulaiman (Sulaiman, 2013), Dr. Hassan Mohamed (Mohamed et

al., 2011) and Ms. Nurul Huda Mohd Saad (Saad et al., 2012) for the corpus. This material is based on work supported by the Universiti Kebangsaan Malaysia (UKM) under Grant No. ERGS/1/2013/ICT1/UKM/3/5.

REFERENCES

Abdul Ghani, R., Zakaria, M. S., & Omar, K. (2009). Jawi-Malay Transliteration. In *International Conference on Electrical Engineering and Informatics 2009 (ICEEI'09)* (Volume:01) (pp. 154–157). Selangor: IEEE. doi:10.1109/ICEEI.2009.5254799

Abdul Rahman, H. (1999). *Panduan menulis dan mengeja Jawi*. Kuala Lumpur: Dewan Bahasa dan Pustaka.

Abdullah, I.-H., Hashim, R. S., & Mohamed Husin, N. (2011). Lexical associations of Malayness in Hikayat Abdullah: A collocational analysis. *Research Journal of Applied Sciences*, *5*(6), 429–433.

Abu Bakar, J. (2008). *Transliterasi Jawi Lama-Jawi Baru berasaskan Grafem (Kajian Kes Hikayat Merong Mahawangsa)* (Unpublished master's thesis). Universiti Kebangsaan Malaysia.

Ahmad, C. W. S. B. C. W., Omar, K., Nasrudin, M. F., Murah, M. Z., & Azmi, S. M. (2013). *Machine transliteration for old Malay manuscript*. In 2nd International Conference on Machine Learning and Computer Science (IMLCS'2013) (pp. 23–26). Kuala Lumpur.

Ahmad, C. W. S. C. W. (2007). *Penterjemah Jawi lama kepada Jawi baru* (Unpublished master's thesis). Universiti Kebangsaan Malaysia.

Atwell, E. (2008). Development of tag sets for part-of-speech tagging. In A. Ludeling & M. Kyto (Eds.), *Corpus linguistics: An international handbook Volume 1* (pp. 501–526). Walter de Gruyter. Retrieved from http://eprints.whiterose.ac.uk/81781/

Azmi, M. S. (2013). *Fitur baharu dari kombinasi geometri segitiga dan pengezonan utk paleografi Jawi digital*.

Bakar, J. A., Omar, K., Nasrudin, M. F., Murah, M. Z., & Ahmad, C. W. S. C. W. (2013). *Implementation of Buckwalter transliteration to Malay corpora*. In 13th International Conference on Intelligent System Design and Applications *(ISDA'13)* (pp. 214–218). Serdang, Selangor: Mirlabs.

Balossi, G. (2014). *A corpus linguistic approach to literary language and characterization: Virginia Woolf's the waves*. John Benjamins Publishing Company.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python* (1st ed.). USA: O'Reilly Media, Inc.

Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. *linguistic data consortium*. Retrieved from https://catalog.ldc.upenn.edu/LDC2002L49

DBP. (2008). *Daftar kata Bahasa Melayu Rumi-Sebutan-Jawi*. Dlm. Dahaman & M. Ahmad (Eds.) (Kedua.). Kuala Lumpur: Dawama.

DBP. (2015). *Pusat Rujukan Persuratan Melayu*. Retrieved from http://prpm.dbp.gov.my/

Diab, M., Hacioglu, K., & Jurafsky, D. (2004). Automatic tagging of Arabic text:From raw text to base phrase chunks. In *HLT-NAACL-Short '04 Proceedings of HLT-NAACL 2004: Short Papers* (pp. 149–152). Stroudsburg, PA, USA: Association for Computational Linguistics.

Diah, N. M., Ismail, M., Ahmad, S., & Abdullah, S. A. S. S. (2010). *Jawi on mobile devices with Jawi WordSearch game application*. In 2010 International Conference on Science and Social Research (CSSR 2010) (pp. 326–329). Kuala Lumpur, Malaysia: IEEE. doi:10.1109/CSSR.2010.5773793

Diah, N. M., Ismail, M., Hami, P. M. A., & Ahmad, S. (2011). *Assisted Jawi-Writing (AJaW) Software for children*. In 2011 IEEE Conference on Open Systems (ICOS2011) (pp. 322–326). Langkawi: IEEE. doi:10.1109/ICOS.2011.6079260

Dukes, K., & Habash, N. (2010). *Morphological annotation of Quranic Arabic*. In Language Resources and Evaluation Conference *(LREC)* (pp. 2530–2536). Valletta, Malta: ELRA.

Habash, N., & Metsky, H. (2008). Automatic learning of morphological variations for handling out-of-vocabulary terms in Urdu-English machine translation. In *Proceedings of the Association for Machine Translation in the Americas (AMTA-08)*. Waikiki, Hawai'i.

Habash, N., Soudi, A., & Buckwalter, T. (2007). On Arabic transliteration. In A. Soudi, A. van den Bosch, & G. Neumann (Eds.), *Arabic computational morphology: Knowledge-based and Empirical Methods*. Springer.

Heryanto, A., Nasrudin, M. F., & Omar, K. (2008). *Offline Jawi Handwritten recognizer using hybrid artificial neural networks and dynamic programming*. In International Symposium on Information Technology, 2008 (ITSim 2008) (Volume:2) (pp. 1–6). Kuala Lumpur: IEEE. doi:10.1109/ITSIM.2008.4631722

Hock, O. Y. (2009). *Kamus dwibahasa*. Petaling Jaya: Pearson Longman.

Irvine, A., Weese, J., & Callison-Burch, C. (2012). Processing informal, romanized Pakistani text messages. In *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012)* (pp. 75–78). Association for Computational Linguistics.

Ismail, K., Yusof, R. J. R., & Jomhari, N. (2010). *A case study of Jawi Editor in the XO-laptop simulated environment*. In 2010 International Conference on User Science and Engineering (i-USEr) (pp. 21–25). Shah Alam: IEEE. doi:10.1109/IUSER.2010.5716716

Knowles, G., & Don, Z. M. (2003). Tagging a corpus of Malay texts, and coping with "syntactic drift." In *Proceedings of the corpus linguistics* (pp. 422–428). Retrieved from http://eprints.lancs.ac.uk/8620/

Leech, G. (2005). Adding linguistic annotation. In M. Wynne (Ed.), *Developing linguistic corpora: A Guide to good practice* (pp. 17–29). Oxford: Oxbow Books for the Arts and Humanities Data Service. Retrieved from http://www.ahds.ac.uk/creating/guides/linguistic-corpora/chapter2.htm

Mohamed, H., Omar, N., & Ab Aziz, M. J. (2011). *Statistical Malay Part-of-Speech (POS) tagger using hidden Markov approach*. In 2011 International Conference on Semantic Technology and Information Retrieval (pp. 231–236). IEEE.

NLTK Project. (2015). *NLTK corpora*. Retrieved from http://www.nltk.org/nltk_data

Noor, N. K. M., Noah, S. A., Aziz, M. J. A., & Hamzah, M. P. (2010). Anaphora resolution of Malay text: Issues and proposed solution model. 2010 International Conference on Asian Language Processing, 174–177. doi:10.1109/IALP.2010.80

Outahajala, M., Zenkouar, L., Benajiba, Y., Rosso, P., & Elirf. (2013). *The development of a fine grained class set for amazigh POS tagging.* In ACS International Conference on. IEEE Computer Systems and Applications (AICCSA) (pp. 1–8). IEEE.

Pasha, A., Al-Badrashiny, M., Diab, M., Kholy, A. El, Eskander, R., Habash, N., … Roth, R. M. (2014). MADAMIRA : A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)* (pp. 1094–1101). Reykjavik, Iceland.

Rahman, S. A., & Omar, N. (2013). Transforming noun phrase structure form into rules to detect compound nouns in Malay sentences. *Journal of ICT*, *12*, 161–173.

Rahman, S. A., Omar, N., & Aziz, M. J. A. (2011). *A fundamental study on detecting head modifier noun phrases in Malay sentence*. In 2011 International Conference on Semantic Technology and Information Retrieval (pp. 255–259). Putrajaya: IEEE. doi:10.1109/STAIR.2011.5995798

Rahman, S. A., Omar, N. B., Mohamed, H., Juzaidin, M., & Aziz, A. (2011). *A synonym contextual-based process for handling word similarity in Malay sentence*.

Redika, R., Omar, K., & Nasrudin, M. F. (2008). *Handwritten Jawi words recognition using hidden Markov models*. In International Symposium on Information Technology, 2008 (ITSim 2008) (Volume:2) (pp. 1–5). Kuala Lumpur: IEEE. doi:10.1109/ITSIM.2008.4631723

Saad, N. H. M., Bakar, J. A., Karim, R. A., Tukiman, N., & Nor, K. M. (2012). *Pembangunan korpus cerpen bertag Bahasa Melayu: Analisis linguistik korpora.* In Research, Invention, Innovation & Design (RIID 2012). Universiti Teknologi MARA Kampus Melaka.

SEAlang Projects. (2011). Retrieved from http://sealang.net/malay/dictionary.htm

Shaalan, K., & Raza, H. (2007). Person name entity recognition for Arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources* (pp. 17–24). Stroudsburg, PA, USA.

Sharum, M. Y., Abdullah, M. T., Sulaiman, M. N., Murad, M. A. A., & Hamzah, Z. A. Z. (2011). *Name extraction for unstructured Malay text*. 2011 IEEE Symposium on Computers & Informatics, 787–791. doi:10.1109/ISCI.2011.5959017

Sulaiman, S. (2013). *Pencantas perkataan Melayu untuk aksara Jawi berasaskan petua*. Bangi: Universiti Kebangsaan Malaysia.

Sulaiman, S., Omar, K., Omar, N., Murah, M. Z., & Abdul Rahman, H. (2014). The effectiveness of a Jawi stemmer for retrieving relevant Malay documents in Jawi characters. *ACM Transactions on Asian Language Information Processing*, *13*(2), 6.

Sulaiman, S., Omar, K., Omar, N., Murah, M. Z., & Rahman, H. A. (2011). A Malay stemmer for Jawi characters. In D. Wang & M. Reynolds (Eds.), *AI 2011: Advances in Artificial Intelligence* (pp. 668–676). Perth, Australia: Springer Berlin / Heidelberg.

Tmshkina, J. (2006). Development of a multilingual parallel corpus and a part-of-speech tagger for Afrikaans. *IFIP International Federation for Information Processing*, *228*, 453–462.

Unicode. (2014). *Unicode*. Retrieved from http://unicode.org

Yonhendri. (2008). *Enjin transliterasi rumi Jawi* (Unpublished master's thesis). Universiti Kebangsaan Malaysia.