Studying the history of the Arabic language: language technology and a large-scale historical corpus

Studying the history of the Arabic language: language technology and a large-scale historical corpus

Cite this article as: Yonatan Belinkov, Alexander Magidow, Alberto Barrón-Cedeño, Avi Shmidman and Maxim Romanov, Studying the history of the Arabic language: language technology and a large-scale historical corpus, Language Resources and Evaluation https://doi.org/10.1007/s10579-019-09460-w

Noname manuscript No.
(will be inserted by the editor)

Studying the History of the Arabic Language

Language Technology and a Large-Scale Historical Corpus

Yonatan Belinkov* · Alexander Magidow* · Alberto Barrón-Cedeño · Avi Shmidman · Maxim Romanov

Received: date / Accepted: date

Abstract Arabic is a widely-spoken language with a long and rich history, but existing corpora and language technology focus mostly on modern Arabic and its varieties. Therefore, studying the history of the language has so far been mostly limited to manual analyses on a small scale. In this work, we present a large-scale historical corpus of the written Arabic language, spanning 1400 years. We describe our efforts to clean and process this corpus using Arabic NLP tools, including the identification of reused text. We study the history of the Arabic language using a novel automatic periodization algorithm, as well as other techniques. Our findings confirm the established division of written Arabic into Modern Standard and Classical Arabic, and confirm other established periodizations, while suggesting that written Arabic may be divisible into still further periods of development.

Keywords Arabic · Corpus · Periodization · Text Reuse · Historical Linguistics

* = Equal contribution

Y. Belinkov
MIT Computer Science and Artificial Intelligence Laboratory and Harvard School of Engineering and Applied Sciences, Cambridge, MA, USA
E-mail: belinkov@mit.edu

A. Magidow
Department of Modern and Classical Languages and Literatures, University of Rhode Island, USA
E-mail: amagidow@uri.edu

A. Barrón-Cedeño
Qatar Computing Research Institute, HBKU, Doha, Qatar
E-mail: albarron@ {hbku.edu.qa | gmail.com}

A. Shmidman
Department of Hebrew Literature, Bar-Ilan University, Israel
Dicta: The Israel Center for Text Analysis
E-mail: shmidman@gmail.com

M. Romanov
Department of History, University of Vienna, Vienna, Austria
E-mail: maxim.romanov@univie.ac.at

1 Introduction

Language is complex and dynamic. It changes across space and time, from generation to generation. New words are introduced and old words go out of use; words acquire new meanings and change their old meanings; and grammatical norms that existed in the past may become obsolete in the future. Language use is partly documented by texts which preserve traces of that change including variations in spelling, prefixes or suffixes that appear or disappear across eras and changes in word meanings.

The Arabic language is no exception to this, but provides a challenge for the historical linguist. Unlike many languages where the standardization of writing was a long and contended process, leaving a trace of that process in the written record, Arabic writing was standardized quite early, while potentially aberrant texts have been subjected to standardizing editorial pressures across many different eras. This created a divergence between the spoken and written languages and while spoken Arabic has continued to evolve, written Arabic was essentially *fossilized* in the 8th century if not earlier. This apparent uniformity is increased by the relatively information-poor Arabic orthography which records only consonants and long vowels (and only in three qualities). Changes like English's Great Vowel Shift would be almost entirely obscured by such an orthography.

There are texts which do diverge from Classical norms, and these have been the subject of significant philological analysis. On the other hand, comparatively little attention has been paid to historical developments in written Arabic which matches the canonical standards. In this paper we seek to develop tools that enable us to determine whether standardized written Arabic is as unchanging and homogeneous as it first appears, or whether we can divide it into separate periods, analogous to those in other languages (e.g., "Old English","Early Modern English"). With 1400 years of written Arabic texts across many different genres and with very subtle differences between eras, this is a task that is incredibly difficult to undertake using traditional philological methods, which may explain the small number of previous studies. For that reason, we seek to develop computational resources and methods to investigate the periodization of standardized written Arabic. Specifically, we seek to answer the following:

1. What computational tools and resources are needed to investigate the periodization of Arabic?
2. Can formal written Arabic be divided into temporally-distinct periods based on linguistic evidence?
3. Are previously proposed periodizations of written Arabic accurate?

After a brief overview of Arabic (Section 2), and a review of previous studies in this area (Section 3), the rest of the paper is divided in two parts. In the first part (Section 4), we describe our efforts to collect, clean, and process OpenITI — a large-scale diachronic corpus of Arabic with approximately 1.5 *G* words. A multi-institutional effort, OpenITI is the largest publicly-available historical corpus of Arabic that we are aware of [59].[1]

In the second part of the paper, we develop computational methods for investigating the history of the Arabic language. In Section 5 we adapt a text reuse

[1] OpenITI is maintainbed at `https://openiti.github.io`. Our pre-processed version is available via `https://doi.org/10.5281/zenodo.2535593`.

algorithm that was previously applied to a relatively small Hebrew/Aramaic corpus in order to identify exact and approximate matches in the large OpenITI corpus. Identifying matches is especially important because many of the documents in the corpus quote and paraphrase large quantities of texts from earlier works, sometimes many centuries earlier. Texts that contain language from very different eras make all forms of historical linguistic analysis more difficult.

In Section 6 we develop a novel data-driven periodization algorithm that is based on word embeddings. Contrary to previous methods, our algorithm captures language use on the level of full corpora or subsets of corpora, rather than being limited to a handful of linguistic features. We apply this algorithm to the OpenITI historical corpus and find well-known as well as new periodizations.

In Section 7 we utilize the corpus for an expert study of Arabic periodization. First, we verify that written Arabic does indeed change remarkably slowly by tracking the lifespan of Arabic words in contrast to English words. We also study several important linguistic phenomena that have so far only been anecdotally analyzed in the literature.

The main contributions of our work are:

- Preprocessing OpenITI, a diverse large-scale historical corpus of Arabic, including morphological segmentation, part-of-speech tags, lemmatization, and syntactic parse trees.
- A complete identification of parallel matches in the corpus based on a novel adaptation of a text reuse algorithm. The algorithm is adapted for Arabic and runs efficiently on the large-scale OpenITI corpus. We are able to identify and remove 292 M words of reused text, nearly 20% of the total corpus.
- A novel periodization algorithm relying on word embeddings, which can be applied to any large-scale historical corpus. We demonstrate its applicability to the Arabic case on OpenITI.
- New insights regarding the history of the Arabic language, that illuminate its development from early times to the modern days. In particular, our computational methods affirm the established periodization of Standard Arabic into Classical and Modern Standard Arabic, and point to new periodizations for Classical Arabic.

2 Linguistic Background

Arabic is a *diglossic* language, meaning there is a single formal language, Standard Arabic (SA), which is used as a language of writing and formal communication.[2] Everyday life is conducted in a divergent set of spoken languages, referred to

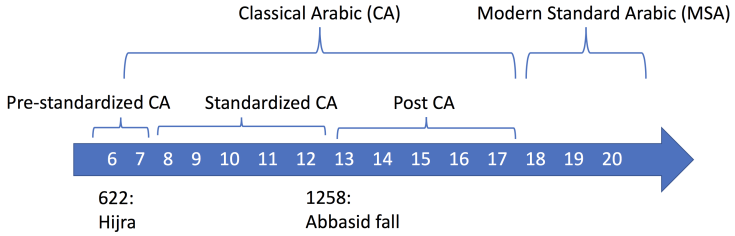
[2] The norm in western Arabist writing is to distinguish between Classical Arabic, the language used prior to the modern era, as opposed to Modern Standard Arabic, the variety developed based on Classical Arabic beginning in the 19th century and continuing to the present. The autochthonous linguistic tradition in the Arab world emphasizes the continuity of the language by referring to the formal, as opposed to the spoken colloquial varieties, as 'al-arabiyya al-fusha' "eloquent Arabic," regardless of era of use. Since we are specifically investigating the periodization of the language, we prefer instead the term Standard(ized) Arabic, which mirrors the term 'al-arabiyya al-fusha' in being agnostic as to era. This term contrasts with written texts in colloquial Arabic, or texts in "Middle Arabic," that is, those which show both colloquial and standard language code-mixed together.

collectively as "colloquial Arabic". Written Arabic in various forms is attested for some time prior to Islam, but the coming of Islam marks the beginning of a vast written tradition. Even prior to Islam, SA[3] was a relatively homogeneous register used for oral literature [24]. However the coming of Islam, dated to 622 CE, when the early Muslim community moved from Mecca to the city of Medina, brought a huge increase in written production. SA was largely standardized even before the 8th century CE, but that is the time period most strongly associated with the establishment of explicit linguistic standards. Oral texts from prior to that time were committed to writing, but were almost certainly edited later to conform more closely to the standard [42]. Such early standardization means that the SA of the 8th century CE is still basically accessible to a reader today — for example, the collection of stories *Kalila wa Dimna* from ca. 750 CE is considered appropriate reading material at the middle school level today, with archaic terms or structures elucidated by footnotes.

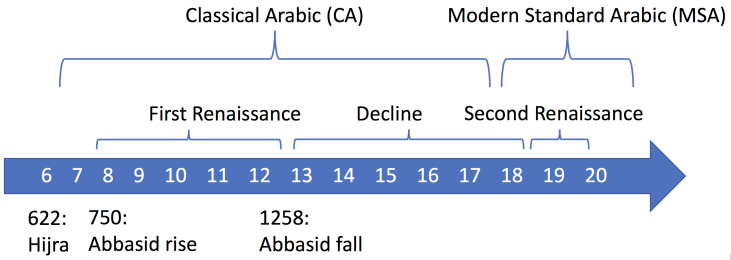
This is not to say that all Arabic writing is homogeneous. There are variants of written Arabic which diverge significantly from SA, even in the pre-modern era: the language of the Ancient North Arabian inscriptions predates the standardization of Arabic, and probably the coming of Islam, and so differs in alphabet, spelling, and lexicon [2]. "Middle Arabic", a term which is not chronological, refers to writings which do not conform completely to the norms of SA, whether they are early Islamic era papyri or Judeo–Arabic letters from the 14th century [72]. These texts are of great use for historical linguists, but a relatively small quantity have been digitized. In the case of Ancient North Arabian texts, they are far too different in genre and vocabulary to be easily comparable to SA texts. By far the largest body of writing is in SA, as this is the language of writing and publishing. Even texts which may have been closer to Middle Arabic at one point were likely corrected by later editors to better conform to the SA style [42].

Impressionistically, there is very little variation between a modern formal text in Arabic and a text from nearly a thousand years ago, if they cover similar topics. Though the literary style of SA varies between genres and eras, the basic orthography, morphology, syntax, and even vocabulary appear to have changed little since that time. This is largely the result of Arabic orthography, with only consonants and long vowels consistently written, short vowels appearing primarily in Quranic verses and very rarely elsewhere. Even changes in consonantal pronunciations might not be reflected in writing, as educated speakers typically write the original (proto)-consonant even if the context demands that it be pronounced with a different, dialectal pronunciation. For example, the pronunciation of the proto-consonant /*q/ varies significantly between dialects, but it is often still written using the ⟨*q*⟩ letter even in dialectal texts, since the pronunciation is understood from context. There may of course be hypercorrections, or confusions when certain consonants have merged, but these mistakes are often removed by the editorial process, so that the print editions most readily available to us show little of that kind of variation. Similarly, the basic morphology of verbal and nominal inflection varies quite little across time, and there is even significant congruence between the inflectional morphology of written Arabic and many modern dialects. There may have been changes and development in syntax, but that is significantly beyond the

[3] A glossary with some definitions and further background appears at the end of this manuscript.



(a) Sub-division of Classical Arabic (CA) into pre-Standardized CA, Standardized CA, and Post-CA, according to Fischer [25].



(b) Periods of renaissance in Standard Arabic (SA), according to Ali [5].

Fig. 1 Two possible accounts of the history of the Arabic language according to philological studies.

scope of this study. The greatest site of variation is in lexis, where words with essentially identical spellings change meaning over time, while new words are coined and others lost.

Therefore, a major question, is whether the apparently homogeneous SA can be divided into eras in the same manner as many European languages, which generally did not experience standardization until late in their literary history. The autochthonous linguistic tradition typically treats SA as a single, undifferentiated language, while western Arabists divide SA into Classical Arabic (CA), the language used in pre-modern texts, and Modern Standard Arabic (MSA). MSA is said to have arisen due to increased contact with the West, starting in the late 18th and early 19th centuries. The largest changes were in lexis, as Western (French, English, German, Italian) terms were adopted or translated, though there may have also been some changes in syntax under the influence of European languages, or even of the spoken colloquial Arabic dialects [50].

Rarely is CA itself divided into other eras, with most variation in CA attributed to register or geography, rather than temporal variation [33, e.g., pp. 38–41]. A small quantity of research supports dividing CA further. There are two slightly different accounts in the literature. Fischer suggests a tri-partite division of pre-Standardized CA, Standardized CA, and Post-CA [25] (Figure 1a). Pre-Standardized Classical Arabic (PSCA) is pre-Islamic and early Islamic, primarily attested in quotations of older texts, and represents a very limited corpus. Standardized Classical Arabic (SCA) fully develops by the 8th century CE, less than two centuries after the coming of Islam, and there are only a small number of minor changes that are claimed to separate it from the Pre-Standardized Era. The period

of Post-Classical Arabic (PCA) is dated to the fall of the Abbasid Empire at the hands of the Mongols (1258 CE), when the power of Arabic-using regimes in Iraq became decentralized, with power shifting elsewhere [58]. Ali suggests a slightly different timeline in which Arabic undergoes two 'renaissances' [5] (Figure 1b). The first begins under the Abbasid Empire in Baghdad (750–1258), witnessing a huge growth in text production and translation of texts from Syriac and Greek. Following the end of the Abbasid Empire empire, Arabic experiences a period of decline until the 19th century renaissance that produces MSA.

These models make slightly different predictions. Fischer's model has SCA coming early and being distinguished primarily from pre-Islamic Arabic and very early texts. Ali's model predicts significant changes during the Abbasid period, particularly an increase in vocabulary during this time. Fischer recognizes PCA as a development of the language, whereas Ali sees it as the end of a renaissance, and hence would not predict rapid change or growth in PCA. Both models treat the fall of the Abbasids in 1258 as a pivotal event in the history of Arabic, but it should be emphasized that this is a political change and only limited linguistic evidence has been shown to separate the two eras [44].

Within this work, we find strong support for the existence of MSA as separate from CA, but within CA the results are less definitive. There is very limited evidence for linguistic changes that occur shortly before the fall of the Abbasids. Other evidence supports the Abbasid renaissance model, with a break between the language of the earliest texts and those of the 4th century AH onward.

3 Related Work

In this section we give an overview of existing Arabic corpora as well as some existing methods for text reuse identification, periodization, and dating.

3.1 Arabic Corpora

Though there has been increasing interest in compiling Arabic corpora in the past decade, very little work has been done on compiling historical corpora reflecting the long history of the Arabic language. Most of the existing corpora focus on modern written Arabic texts, particularly online news media, although there are a growing number of corpora which feature written and to a lesser degree spoken material from Arabic dialects. We mention here several relevant corpora and refer to other surveys for more details [3,4,67,73].

To date, only a small number of diachronically oriented corpora of Arabic have been produced and made available. The King Saud University Corpus of Classical Arabic (KSUCCA) [6][4] consists of approximately 50.6 M words from the first 4 Islamic centuries. It has been morphologically analyzed with the MADA tool [28, 29]. Almost all of the texts are derived from the Shamela website which is a major source of texts for OpenITI.[5] Text metadata is given by century, so more granular buckets are not possible in the current state of this corpus. The Historical Arabic

[4] `http://ksucorpus.ksu.edu.sa`

[5] `http://shamela.ws`

Corpus (HAC) [32] has about 45 M words from diverse time periods, with text data given by century, as well as automatic part-of-speech tagging information. Other Classical Arabic corpora that are worth mentioning include a 5 M word corpus [23], which does not seem to be publicly available, another 2.5 M word corpus [56],[6] and Tashkeela, a 76 M word corpus of texts from the Shamela website [76].[7] These corpora are either small or lack high-quality temporal metadata.

Finally, a few large corpora are available only via online search interfaces: KACST Arabic Corpus [4] has more than 700 M words, including around 16 M words from the beginning of the Islamic era. The Leeds Arabic Internet Corpus[8] and the International Corpus of Arabic[9] contain 300 M and 100 M words, respectively, but they include mostly modern texts. The well-known ArabiCorpus[10] has more than 170 M words from diverse periods of time, and arTenTen [7] is a 5.8 G word Web corpus, with a sub-corpus of 115 M words available through Sketch Engine [38]. There is also CLAUDia [74], another Shamela-based corpus, but with added genre metadata; however, only a subset appears to be accessible via a Web interface.[11] While these corpora are very large and may contain texts from different periods, they are not directly accessible and lack sufficient diachronic information.

In contrast to previous resources, our corpus has fine-grained time information, it covers most of the history of the written Arabic language, and it is available for developing NLP applications or supporting digital humanities projects (cf. Section 4).

3.2 Identification of Text Reuse

A popular approach to text reuse addressed the problem in the context of domains such as newspaper texts and law bills [16,68,71]. A standard approach to approximate-matching tasks is the use of edit-distance measures such as Levenshtein Distance (e.g., [63]); however, such an approach is not efficient, particularly given a corpus of this size. More successful and efficient models rely on either word- or character-level n-grams to align similar chunks of text [37,49]. Documents are broken down into overlapping sequences of relatively short n-grams (~ 4 for words; ~ 16 for characters) and the resulting hashes are either indexed for search or compared pairwise in order to find collisions.

Some text reuse detection models have been open-sourced, such as `passim`.[12] A recent international challenge focused on text reuse detection in Arabic [10], but the reuse cases were artificially generated in order to allow for an objective evaluation. The approaches that addressed the task were mostly based on n-grams comparison as well, with some Arabic-specific preprocessing. An especially interesting study of real text reuse is [75], which detected quotations in the CLAUDia

[6] `http://www.RDI-eg.com/RDI/TrainingData`
[7] `https://sourceforge.net/projects/tashkeela`
[8] `http://corpus.leeds.ac.uk/internet.html`
[9] `http://www.bibalex.org/ica/en/About.aspx`
[10] `http://arabicorpus.byu.edu`
[11] `http://arabiccorpus.com/index.htm`
[12] Available at `https://github.com/dasmiq/passim`; previously applied to Arabic texts: `http://kitab-project.org/text-reuse-methods/`.

corpus (cf. Section 3.1) and built a network of documents based on metadata and quoted texts. However, their method focuses on long verbatim quotations, whereas we are interested in approximately-matching parallel passages with possible variations. Instead, we follow a recent approach for finding parallel passages across a large Hebrew/Aramaic corpus [66], adapt it to the Arabic language, and scale it up to handle the large corpus.

We refer the reader to [15, 43, 68] for a broader overview of text reuse detection models and their applications.

3.3 Periodization, Text Dating, and Language Change

Most approaches to periodization, qualitative and quantitative, have either assumed standard periodizations proposed by traditional linguistics, or worked with pre-determined temporal bins (e.g. decades or centuries). Some recent works use clustering algorithms to determine more natural periodizations, but these approaches require pre-selected variables for classification. For example, in a study on automatic clustering of English, frequencies of get-passives and verb conjugation suffixes *-(e)th* and *-(e)s* were used as input for the clustering algorithm [26]. In a study on Chinese, the variables were already selected with an awareness of the history of Chinese morphosyntax, and the variables themselves were encoded into the annotated corpus [34]. Although these methods are promising, they require an existing sense of meaningful variables which vary diachronically, whereas we seek to periodize language without subscribing to pre-defined variables.

There is a fairly large body of work on text dating, especially using clues like time expressions, but also various other features [13, 17, 51, 54]. Previous research operated at different granularity levels and algorithmic methods, including pairwise learning-to-rank, multi-class support vector machines, and language models [51, 54, 36]. These methods aim to learn to assign dates to undated documents. Our main motivation in this paper is different: given a corpus of dated texts, we seek to find a division into historical time periods.

Another line of work has applied computational methods for studying diachronic language change. These methods learn the context in which words appear in large corpora, and define semantic change as a change to that context. Earlier attempts used this principle to detect meaning change in 19th century British English [62], or in a more modern corpus [70]. However, they tested semantic change for a very small set of words. In contrast, recent work has extended the scope of such analysis, and is now able to automatically detect words with significant semantic change for the entire vocabulary, building on word embeddings learned from large raw texts [22, 31, 39, 40]. We develop a periodization algorithm based on word embeddings to automatically cluster periods of similar language use.

4 Corpus Construction

In this section we describe the OpenITI corpus as well as the preprocessing we carried out on it in order to perform our computational linguistic analyses.

	Words	Documents
OpenITI full	1.5 G	7,144
OpenITI core	725 M	4,322

Source	Documents
Shamela	2,375
JK	1,106
Shia library	823
Others	8

Table 1 Statistics on the OpenITI corpus showing the number of words and documents in OpenITI full vs. OpenITI core (left), and the distribution of sources in OpenITI core (right).

4.1 Text Collection

The corpus we present in this work is the Arabic portion of the works collected by the Open Islamicate Texts Initiative (OpenITI), an on-going effort to collect texts in Arabic, Persian, and other languages.[13] The documents are collected from freely available editions of primarily religious and literary texts from different time periods. The main sources of these texts include Al-Maktaba Al-Shamela,[14] referred to here as Shamela, the Shia online library,[15] and Al-Jami' Al-Kabir (JK).[16] The documents were converted into a unified format and organized into a machine-readable corpus, which is now openly available.[17] Accompanying metadata information includes standardized author names, text title and author date of death.[18] Though Shamela includes text genre data, it is not very reliable and so this was not retained. We use the author date of death as the document date throughout this work. Author lifespans average around 70 years, so there is a lag surrounding all of the data that needs to be taken into consideration during analysis [60, p. 239]. All dates in the corpus are based on the Islamic calendar, which begins in 622 and which uses lunar years. Where reasonable, we have converted these dates to Gregorian years, but where they would produce unrounded bins and boundaries, we have retained the Islamic century numbering.

The current OpenITI corpus actually retains all duplicate texts. We refer to this as OpenITI full. De-duplication has systematically chosen an instantiated text for each unique work from the source corpora. We refer to this as OpenITI core. Table 1 shows some statistics of the two versions. After de-duplication, OpenITI core has 1106 texts (26%) from JK, 2375 (55%) from Shamela, and 823 (19%) from the Shia library; 8 texts are from elsewhere. The de-duplication was conducted in a semi-automatic manner.[19]

Table 2 shows detailed statistics on the OpenITI full corpus of texts, including the distribution of texts, sentences,[20] and words over time as reflected in the corpus. As illustrated in Figure 2, we observe three major jumps in the number

[13] `http://iti-corpus.github.io`

[14] "The Complete Library", `http://shamela.ws`

[15] `http://shiaonlinelibrary.com`

[16] "The Great Collection". Not available online; it was published on an external hard-drive.

[17] `https://github.com/OpenITI`

[18] Standardization of author names and text titles is a difficult task for Arabic and was performed using a combination of digital tools (`https://github.com/maximromanov/DuplAway`) and manual inspection.

[19] More details are available at: `https://maximromanov.github.io/OpenITI/`.

[20] Based on the sentence splits created via preprocessing (Section 4.2).

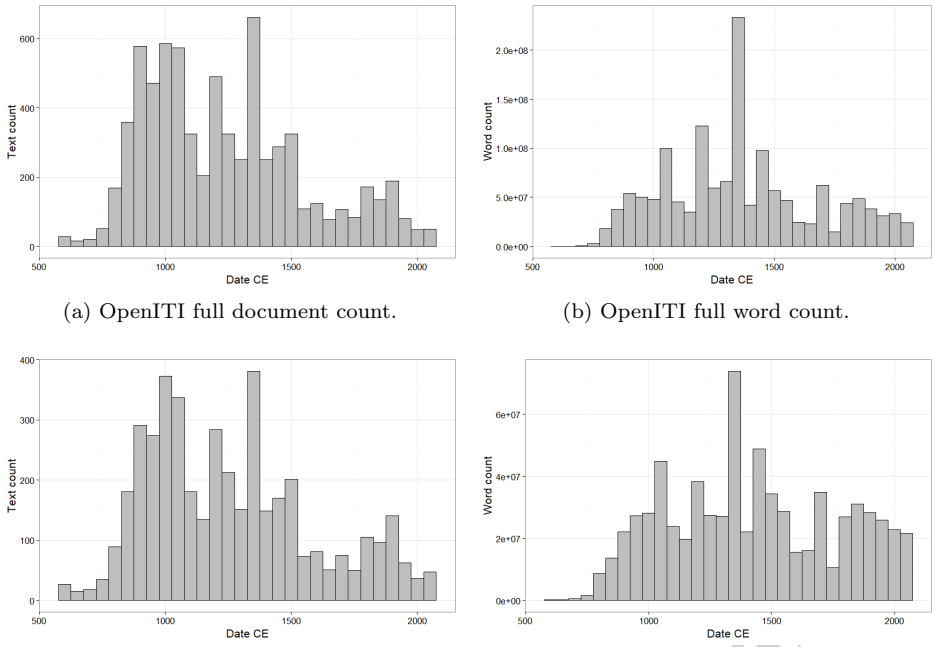
Period (AH)	Texts	Sentences	Words	Period	Texts	Sentences	Words
1–50	43	33K	462K	751–800	407	3,980K	112M
51–100	17	33K	546K	801–850	226	2,105K	47M
101–150	48	122K	3M	851–900	240	3,396K	98M
151–200	120	549K	11M	901–950	288	2,248K	55M
201–250	325	1,487K	39M	951–1000	122	1,536K	45M
251–300	504	1,779K	39M	1001–1050	112	858K	29M
301–350	495	2,539K	65M	1051–1100	103	1,533K	37M
351–400	568	2,850K	59M	1101–1150	82	1,713K	47M
401–450	458	2,075K	46M	1151–1200	81	482K	14M
451–500	481	3,350K	102M	1201–1250	211	2,666K	66M
501–550	266	1,919K	41M	1251–1300	100	1,004K	37M
551–600	443	3,392K	101M	1301–1350	201	1,713K	36M
601–650	291	2,361K	67M	1351–1400	53	1,537K	34M
651–700	313	2,216K	63M	1401–1450	90	2,714K	56M
701–750	456	9,597K	186M	**Total**	7144	62M	1,537M

Table 2 Number of texts, sentences, and words in each 50-year time period in the OpenITI full corpus. AH refers to the Islamic calendar period.

of documents: one in the early period, ca, 800 CE, one in the middle period, ca. 1400 CE, and one in the modern period, after 1800 CE. These mostly correspond to increases in the number of words as well.

As is evident from the statistics, some time periods are more represented in the corpus than others. As in most historical corpora, guaranteeing representativeness is often not possible [14]. Therefore, we decided to keep all available texts rather than artificially selecting a balanced sub-set of them, in an effort to be as comprehensive as possible.

Comparison with Shamela Initial findings of this work have been reported in the 2016 Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH) [9]. That work made use of texts from the Shamela text collection. This text collection has good coverage of pre-modern texts, and an excellent section of modern texts. However, many foundational texts, particularly texts which are more literary than religious, were absent from that collection. The current work is based on the Open Islamicate Texts Initiative, which integrates texts from a variety of text collections — including semi-automated selection of the ideal copy of each text where multiple copies are present. This has the advantage of gathering a greater coverage of the pre-modern era, such that OpenITI has 500 more texts from the period before 1800 (15 M more words). However, less effort has been made thus far to reintegrate modern texts into OpenITI, so Shamela contains 860 more modern texts than OpenITI, totaling 77 M words. Ongoing initiatives exist to reintegrate the Shamela texts into OpenITI and to increase the number of texts, specifically texts from the early modern era (1800-1900) as that period is when MSA is said to have started developing.



(a) OpenITI full document count.

(b) OpenITI full word count.

(c) OpenITI core document count.

(d) OpenITI core word count.

Fig. 2 Document and word counts per century in the OpenITI corpus versions.

4.2 Preprocessing

We used a combination of OpenNLP[21] and the *Farasa* toolkit [1][22] to pre-process the corpus. Farasa is a popular toolkit for Arabic NLP which was found by multiple studies [19] to perform comparably to MADAMIRA [1,53,20], another popular toolkit. The two toolkits were developed mainly for MSA, but there is no known comprehensive evaluation of them on historical Arabic.[23] We provide a small manual evaluation of Farasa on our corpus below, as well as compare the preprocessing results of Farasa and MADAMIRA quantitatively. Our evaluation demonstrates that there is little difference in their performance, and that both are viable choices for preprocessing historical Arabic.

Next we describe our preprocessing pipeline.[24]

Sentence splitting. This is the only component not available in Farasa. Punctuation is inconsistently used in Arabic and a stretch of text between periods may contain many sentences. We used OpenNLP and trained the sentence split-

[21] `https://opennlp.apache.org`

[22] `http://farasa.qcri.org`

[23] One study evaluated the two tools on diacritization of Classical Arabic, and found Farasa to perform much better than MADAMIRA [52].

[24] The code to perform this pre-processing is available at `https://github.com/albarron/AraProc`.

ting model on 5 K sentences from the AQMAR Arabic Wikipedia Supersense corpus [64] and NIST's MT06 corpus.[25]

Morphological segmentation. The segmenter breaks words into their underlying morphemes. Farasa's segmenter uses SVM^{rank} [35] with a linear kernel to determine the best segmentation for each word.

Lemmatization. The lemmatiser is a rule-based system. First, a word is looked up in a word–{diacritization, lemma} dictionary and the lemma of the most frequent diacritized word is returned if located. Otherwise, the Farasa segmenter is applied to extract the word's morphemes and the first non-prefix morpheme is looked up again in the dictionary. If found, the mapped lemma is returned; otherwise, the morpheme is returned.

Part-of-speech tagging. The POS tagger uses the simplified PATB tagset proposed by [18]. It attempts to find the optimal tag for each morpheme produced by the segmenter, as well as determining the gender (masculine or feminine) and number for nouns and adjectives (singular, dual, or plural). The POS tagger uses SVM^{Rank} to find the best tag for each morpheme as well.

Constituency parsing. This is a re-implementation of the Epic parser [30], which performed best at SPMRL 2013 [11]. It uses a conditional random fields model trained on features derived from the POS tagger.

We utilize the different output formats produced by the preprocessing throughout this work, except for the parse trees. We still make the trees available to facilitate future syntactic work on the history of Arabic.

Qualitative evaluation Farasa is tuned for the news domain and for MSA; still, "it can handle other genres along with classical and dialectal Arabic" [61]. We performed a qualitative analysis of the Farasa results to ensure that it still performed well with Classical Arabic texts. We analyzed the results of the segmentation, POS-tagging, and lemmatization in a set of texts chosen based on genre and date.[26] In all of the texts, the quality of the Farasa results was high, with relatively few errors in the segmentation and lemmatization, even for words that are obsolete or Classical. In a small sample chosen for manual quantitative analysis, comprising a total of 507 words in 4 texts, there were 41 mistakes of all kinds, with Farasa correctly segmenting 98.82%, lemmatizing 98.62% and POS-tagging 94.28% of words in this small sample. Classical coverage appears adequate — for example, in a pre-Islamic poem,[27] the lemmatizer correctly derived the lemma *A$Abp* 'mixed group of people' from the plural form *A$A}b*, though this word is almost never used in modern language. This suggests that Farasa's lexicon has an adequate coverage of CA vocabulary in spite of being designed for MSA. On the other hand, the POS tagging occasionally produced incorrect gender information, and mislabeled verbs as nouns, though the lemmatization produced the correct form in most cases, and this is an ongoing issue even in MSA texts. One instance of a specifically CA

[25] `https://www.nist.gov/programs-projects/machine-translation`

[26] Texts tested were the following (the first digits are the AH dates):
`0001NabighaDhubyani.Diwan.JK007511` (pre-Islamic poetry),
`0680IbnSabuni.TakmilaIkmalIkmal.JK000884` (Hadith collection),
`0685IbnSacidMaghribi.Jughrafiya.Shamela0000463` (geography),
`1405CaliShahrudi.Mustadrakat.Shia002984Vols` (modern devotional).

[27] OpenITI document `0001NabighaDhubyani.Diwan.JK007511-ara1`.

structure that posed a difficulty for Farasa was the energetic form *l-yltms-n* 'he will seek out', which Farasa treated as a single word in all analyses. The energetic is largely absent from MSA, and marginal in its use in CA. However, Farasa had some difficulty segmenting verbs, especially present-tense verbs, from conjunctions and other prefixes, so this may be a related issue. The output of Farasa is generally useful for a variety of research types, but is specifically useful for concordancing and word frequency counts, since removal of clitics and identification of lemmas poses a significant problem in Arabic computational linguistics.

Farasa vs. MADAMIRA Both Farasa and MADAMIRA are high-quality toolkits for Arabic NLP. For periodization, we do not expect to see a large impact of the choice of tool, as we are interested in high-level trends, rather than obtaining a few more points in performance. Nevertheless, we provide here quantitative and qualitative results showing that the differences between the tools are minor and insignificant for our purposes.

To test this, we ran the two toolkits on 58 texts from our corpus, one from each 25 year time period, and compared their results by computing the character edit distance for each preprocessed sentence. When comparing morphological segmentation, the average edit distance was about 20.1 edit operations, corresponding to 11.2% of the characters (when dividing by the total sentence length). After accounting for different normalization schemes, the average edit distance was about 8.9 operations, corresponding to just 5.9% of the characters. We also ran a similar evaluation for lemmatization and found even smaller differences between Farasa and MADAMIRA. On lemmatized texts, the average edit distance was 6.8 edit operations, which correspond to 5.4% of the characters.

These results indicate that the choice of toolkits for preprocessing would not have a major effect on our analysis. More importantly, such small differences do not have a high impact in periodization, where we are concerned with global, overall trends, rather than with improving the state-of-the-art by some small fraction.

5 Text Reuse

In order to identify instances of text reuse within the corpus, we adapt SKP [66], a recent approach designed to efficiently find approximately-parallel passages across a large Hebrew/Aramaic corpus. This method is appealing to use in our case due to the similarity between Arabic, Aramaic and Hebrew: all are languages of Semitic origin with a high morpheme-per-word ratio. First we detail two modifications that we had to apply before running the algorithm in order to handle the nature of the OpenITI corpus. We refer to this enhanced algorithm as SKP-Ar. The whole procedure is summarized in Algorithm 1.

5.1 Identification of Boilerplate Passages

Before running the processor-intensive algorithm to identify approximate matches within the corpus, we isolate "boilerplate" passages that recur verbatim dozens, hundreds, or even thousands of times within the corpus (e.g., Quranic verses are quoted extremely frequently). A common genre in the corpus is Hadith collections,

Prophetic reports, which must be preceded with a chain of transmission from the person who heard it directly from the Prophet through the main person who transmitted it to the latter transmitter. Thus, both the chain of transmission and the quotation itself tend to be widely repeated. Others are simply frequently mentioned anecdotes or sayings.

To identify boilerplate passages, overlapping phrases of length 20 are extracted and counted from the whole corpus (line 3) and those phrases which have appeared 25 or more times verbatim are marked (line 9). Phrases are clustered and conflated if they appear overlapped or juxtaposed (allowing for a gap of up to 10 words between the 20-gram segments) and the resulting text fragments are marked as boilerplate passages. These text fragments are ignored by the subsequent stages of the approximate-matching algorithm, allowing them to focus on the more meaningful parts of the text, without getting bogged down in these commonly-recurring exact matches.

Appendix B shows examples of boilerplate passages identified in this step.

5.2 Identification of Extremely Frequent Short Phrases

The second step involves the identification of frequently recurring 4-grams. The core of the SKP algorithm involves hashing and indexing 4-grams throughout the whole corpus. This works well for Hebrew and Aramaic, since frequently-recurring Hebrew and Aramaic formulaic phrases tend to be limited to two (sometimes three) words, and thus 4-grams prove to be effective units in processing the text. However, within the OpenITI corpus, we find many 4-grams that recur repeatedly. These are often blessings upon the Prophet Muhammad, which are extremely frequent (e.g., `SlY Allh Elyh wslm`, "peace be upon him", recurs over 5 M times). The prevalence of such phrases could render SKP [66] highly ineffective.

In order to solve this problem, we identify the top 35 K frequently-recurring 4-grams in the corpus (function `FindFrequentPhrases`). We assign each of these phrases a unique 16-bit hash, and henceforth treat each of those phrases as a single word unit, each one with its own unique hash. When launched on the OpenITI corpus, the selected phrases appeared between 515 and 5 M times each.

5.3 Identification of Approximate Matches

The function `IdentifyReuse` details the main steps in the SKP-Ar algorithm. As an intermediate step towards finding long parallel passages, we first want to identify all cases of approximately-matching short phrases between the documents. For our purposes, we regard an approximately-matched short phrase as cases of phrases of length five in which four out of the five words are nearly identical. We define a skipgram as any four-word subset of a five-word string. For every 4-out-of-5 skipgram within the text (excluding the boilerplate material, as described above), we assign a 64-bit hash, comprised of the two least-frequent letters of each of the four words (line 27 in Algorithm 1). An initial pass of the program reviews the entire corpus and builds a character frequency table of all characters classified as Arabic characters by the Unicode specification. The corpus includes 131 such characters; thus any two-letter combination fits easily into 16-bits, and also allows

input : $Texts$: a list of T documents, ordered chronologically
output: $BoilerPlates$: the boiler plate fragments
$ReusedFragments$: the reused fragments

1 **Function** `FindBoilerPlates`($Texts$)
2 Initialize dictionary $Phrases(phrase, counter)$
3 **for** $i \leftarrow 1$ **to** T **do**
4 **for** $j \leftarrow 1$ **to** $J_i \mid |j| = 20, step = 1$ **do**
5 $Phrases \leftarrow j$
6 **end**
7 **end**
8 $Phrases \leftarrow [\forall phrase \in Phrases \mid freq(phrase) \geq 25]$
9 $BoilerPlates \leftarrow$ `Conflate`($Phrases$)
10 **return** $Boilerplates$
11 **Function** `FindFrequentPhrases`($Texts$)
12 Initialize dictionary $Phrases(phrase, counter)$
13 **for** $i \leftarrow 1$ **to** T **do**
14 **for** $j \leftarrow 1$ **to** $J_i \mid |j| = 4, step = 1$ **do**
15 $Phrases \leftarrow j$
16 **end**
17 **end**
18 $Phrases \leftarrow$ `SortByFrequency`($Phrases$)
19 $ExtremelyFrequentPhrases \leftarrow$ `Top`($Phrases, 35000$)
20 **return** $ExtremelyFrequentPhrases$
21 **Function** `IdentifyReuse`($Texts$)
22 $BoilerPlates \leftarrow$ `FindBoilerPlates`($Texts$)
23 $Texts' \leftarrow [Texts \setminus BoilerPlates]$
24 $XtremFreqPhrases \leftarrow$ `FindFrequentPhrases`($Texts'$)
25 $Text'' \leftarrow$ `ConflateFreqPhrasesIntoSingleWords`($Texts', XtremFreqPhrases$)
26 $SkipGrams \leftarrow$ `ComputeSkipGrams`($Texts''$)
27 $SkipGramsHashes \leftarrow$ `HashSkipGrams`($SkipGrams$)
28 Initialize list $ReusedFragments$
29 **for** $i \leftarrow 1$ **to** T **do**
30 **for** $j \leftarrow 1$ **to** T **do**
31 $BaseMatches \leftarrow$ `GetMatchingHashes`($T_i, T_j, SkipGramHashes$)
32 $FullMatches \leftarrow$ `ExtendMatches`($BaseMatches$)
33 `Append`($ReusedFragments, FullMatches$)
34 **end**
35 **end**
36 **return** *BoilerPlates, ReusedFragments*

Algorithm 1: The SKP-Ar algorithm for text reuse identification.

space for the 35 K unique hashes for the frequently-recurring 4-gram phrases detailed in the previous step. The determination of the two least-frequent letters in a given word is also based upon this initial review of the character inventory within the corpus. This method conveniently facilitates approximate matches. The use of 4-out-of-5 skipgrams allows passages to match up even though a given word may be subtracted, added, or replaced within the unit. Similarly, the two-letter word hashing allows words to be considered equal despite differences in prefixes, suffixes, or *matres lectionis*.

Now, for any given document (the "base document"), we tabulate all cases in which one of its skipgram hashes matches a skipgram hash from another document (the "target document") (line 31). We wish to identify cases in which multiple skipgram matches are in close proximity with one another to form a passage of

substantial length. To do so, we generate a two-dimensional graph, wherein each skipgram match is plotted on one axis according to the starting word position in the base text, and on the other axis according to the starting word position in the target text, similar to a dotplot [8,27]. We are interested in the cases in which multiple skipgram matches cluster on a more-or-less diagonal line on the graph. To efficiently find such cases, we bin the skipgrams based upon the difference between their two coordinates. We review the bins which contain multiple skipgrams and consider whether those skipgrams can cluster together to form a match containing 16 or more identical words, allowing up to 3 non-matching word positions in between any two matching skipgrams (line 32).

The use of 4-out-of-5 64-bit hashes casts a rather wide net from the start, wherein many identical skipgrams actually point to very different phrases. However, this is compensated by the requirement to have a series of adjacent matching skipgrams. As the number of adjacent skipgrams cluster together, the number of false positives drops progressively lower. In our case, where we require a series of matching skipgrams which match up at a minimum of 16 word positions, we find that the resulting passages are virtually always legitimate cases of text reuse.

5.4 Results

We ran the SKP-Ar algorithm on the entire 1.5 G word OpenITI corpus. The algorithm first isolated 230,530 unique (though possibly overlapping) frequently-occurring 20-word phrases (boilerplate strings). Each phrase occurs at least 25 times within the corpus, with the most frequent phrase occuring 4,495 times. In total, we mark 28,491,859 words out of the total 1.5 G words as boilerplate text. After eliminating the boilerplate text, the approximate-matching algorithm returned 76 M pairwise matches, with an average length of 46.7 ± 249 words per match. The process took 48 hours, running in parallel on 64 CPUs.

Given that the texts are dated by author's date of death, we allowed some relaxation in finding reused text chunks. We counted only cases in which at least 50 years elapsed between the dates of the earlier and later documents. After this filter, we are left with 57 M pairwise matches, with an average length of 31.58 ± 35.12. Note that this filter also eliminates duplicate texts that have the same date.

Finally, we calculated the total number of reused words within the corpus, leveraging both the set of boilerplate phrases as well as the set of approximately-matching passages. After applying the 50-year filter, we were left with a total of 292 M reused words within the corpus.

Efforts have been carried out to standardize the evaluation of text reuse models [55], but they rely on artificially-generated cases of reuse. Evaluating the performance of a model when applied to a real-life corpus, such as OpenITI, remains a difficult task. Most challenging is evaluating the exhaustiveness of matching. We expect to find extremely important texts to be quoted the most, so the number of matches by text are indicative of whether matches are being correctly identified. Indeed, the largest numbers of boilerplate quotations, i.e. quotations repeated 25 times or more, come from the Quran and major works of religious exegesis and historical works, all of which we expect to see widely quoted. Manual examination of the approximate pairwise matches is also positive — though not quantifiable, rapid scrolling through the lists of results provides easy visual identification of

the similarity of the matches, and closer evaluation reveals slight variations in the quotations, often slight reformulation of the phrasing. Visually checking several lists of matches by file did not reveal any obviously flawed matches.

6 Automatic Periodization

With access to a diachronic corpus, we are able to investigate the linguistic developments which have been claimed to characterize the different stages of CA. We developed an automatic algorithm for dividing a historical text corpus into time periods. We then applied it to the OpenITI corpus and analyzed the obtained results.[28]

It should be noted that typically, language periodization is based on a broad variety of factors and changes at all level of the language, from phonology to morphology, syntax and semantics. As described previously, however, our corpus provides no meaningful access to information about phonetics, since Arabic script does not typically reflect phonetic variation, especially in the edited, print editions that form the basis of our corpus. Inflectional morphological differences are nearly non-existent between eras, though we are able to investigate the history of some derivational morphemes. Syntax may be a rich area for future investigation, but would require significant work to verify quality of automatic constituency parsing, in addition to development of a comparison algorithm. This leaves us primarily with lexical-semantic change, which is both well reflected by our corpus and is the primary area of previous philological investigation. Future research can try to integrate our findings with other areas of linguistic change.

6.1 Word-Embedding-based Neighbor Clustering

Given the nature of language change, we note that the language in two consecutive time periods should in principle be more similar than the language in two remote time periods. Therefore, we apply a chronologically-constrained hierarchical clustering algorithm that is only allowed to merge consecutive time periods. The core of the algorithm is based on a word-embedding function and a distance function. The word-embedding function takes a text document and generates a word-embedding matrix. The distance function takes two word-embedding matrices, corresponding to two time periods, and computes the distance between them. Below we discuss specific instantiations of these functions. Our algorithm can be seen as a word-embedding-based variant of the Variability-based Neighbor Clustering (VNC) algorithm for periodization [26], where we replace the measure of variability by a distance measure based on word-embedding matrices. Word embeddings are attractive to use for this purpose because they provide a soft notion of language use, with similar words having similar vectors in the word embedding space [47]. We name our periodization algorithm Word-Embedding-based Neighbor Clustering (WENC).

We assume a collection of texts, $\mathcal{T}$, with known dates. We first bin the texts into initial time periods $\mathcal{P} = \{P_1, ..., P_{|\mathcal{P}|}\}$, that are ordered chronologically (e.g.,

[28] The periodization code is available at `https://github.com/boknilev/periodization`.

```
   input : Texts: a list of T documents, ordered chronologically
   output: MergedPairs: the merged clusters
           MergedDistances: the corresponding distances
 1 Function Periodize(Texts)
 2     Initialize list Models
 3     for i ← 1 to T do
 4         Models[i] ← TrainWordEmbModel(Texts[i])
 5     end
 6     Initialize lists MergedPairs, MergedDistances
 7     while |Texts| > 1 do
 8         BestPair, BestDist, Texts, Models ← FindBestMerge(Texts, Models)
 9         Append(MergedPairs, BestPair)
10         Append(MergedDistances, BestDistance)
11     end
12     return MergedPairs, MergedDistances
13 Function FindBestMerge(Texts, Models)
14     Initialize list Distances
15     for i ← 1 to |Texts| − 1 do
16         Distances[i] ← ComputeDistance(Texts[i], Texts[i + 1])
17     end
18     BestPair, BestDistance ← ArgMin(Distances)
19     MergedText ← Concat(Texts[BestPair])
20     MergedModel ← TrainWordEmbModel(MergedText)
21     Texts ← UpdateList(Texts, MergedText)
22     Models ← UpdateList(Models, MergedModel)
23     return BestPair, BestDistance, Texts, Models
```

Algorithm 2: WENC: word-embedding-based neighbor clustering for automatic periodization.

centuries). That is, for each $i < j$, all the texts in P_i are dated earlier than all the texts in P_j. The texts in each time period are concatenated into documents that are input to the periodization algorithm (see Algorithm 2). We start by training initial word embedding models (line 4). Then, we iteratively look for the next best possible merge of time periods until there are no more time periods to merge (line 8). At each iteration we record the best merges and distances (lines 9-10).

The algorithm utilizes a function `FindBestMerge` that takes a collection of documents and their corresponding word embedding models and computes the distances between each consecutive pair of documents (line 16). It then finds the best pair (line 18), concatenates the two documents (line 19), trains a word embedding model on the new concatenated document (line 20), and updates the list of texts and models (lines 21-22).

6.1.1 Word Embeddings and a Distance Measure

The periodization algorithm relies on training word embeddings on texts in each time period (line 16 in Algorithm 2). We obtain the word embeddings using `Word2Vec` [45,46,48] as implemented in `gensim` [57]. Specifically, we train the CBOW algorithm with negative sampling and the following default settings defined in `gensim`: word embedding dimensionality of 100, 5 negative samples, and a window size of 5 words.

Given two word embedding matrices, W_1 and W_2, trained on different corpora, we need to define a distance measure between them. One option could be to directly

calculate the distance with respect to some norm:

$$\texttt{ComputeDistance}(W_1, W_2) = ||W_1 - W_2|| \tag{1}$$

However, since the two word embedding matrices are trained independently, we have no guarantee that this distance measure would yield meaningful results. Moreover, the stochastic nature of the word embedding training algorithm precludes a direct comparison of words from different embedding models. To avoid this problem, we follow [31] and align the two matrices using orthogonal Procrustes:

$$\texttt{ComputeDistance}(W_1, W_2) = \min_{Q:Q^TQ=I} ||QW_1 - W_2||_F \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm of matrix $\cdot$ and Q is an orthogonal matrix that rotates the word embedding matrix W_1 towards W_2. The solution to this minimization problem is given by the best rotation matrix and can be found with singular value decomposition (SVD) [65]:

$$R = \underset{Q:Q^TQ=I}{\arg\min} ||QW_1 - W_2||_F = UV^T \tag{3}$$

where $W_2W_1^T = U\Sigma V^T$ is the SVD decomposition.

6.2 Experiments

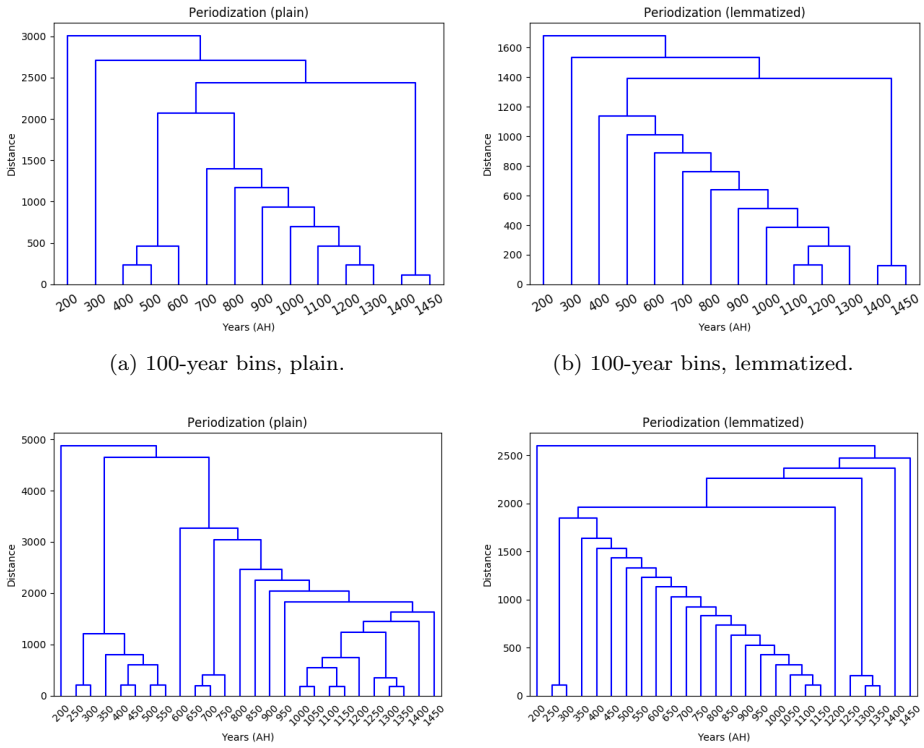
For the periodization experiments, we consider two possible initial time divisions: 50 and 100 year bins.[29] Table 2 shows the number of texts, sentences, and words, for each 50-year bin. The 100-year bins are simply a concatenation of each two consecutive 50-year bins. We find that the very early time periods contain much less text, so we merge the bins for the first 200 years in all of the experiments (2 and 4 bins are merged in the case of the 100-year and 50-year bins, respectively). In the following, we investigate the effect of preprocessing on periodization, by considering two input formats for the periodization algorithm: plain text and lemmas. Working with lemmas allows the periodization algorithm to focus more on lexical properties rather than surface forms. However, the morphological lemmatization performed by Farasa (Section 4.2) is an automatic process that may produce errors, so we also run the periodization algorithm on plain text.

We also investigate the effect of text reuse by comparing the periodization results on the full corpus to running the same algorithm on a version of the corpus where reused text chunks were removed.

In all cases, we run the WENC periodization algorithm (Algorithm 2), record the hierarchical merges and their distances, and plot the results in dendrograms.

Figures 3a and 3b show the results of running WENC on 100-year time periods, using plain and lemmatized texts, respectively. In both cases, we see a split into three main periods: early period until 200/300 AH, middle period from 200/300 AH to around 1300 AH, and a late period from that time to modern days. In the case of 50-year bins (Figures 3c and 3d), we can observe a more fine-grained periodization. The two figures are very different: the periodization based on plain

[29] Since the OpenITI corpus is dated by Islamic years, we use 50/100 Islamic year bins.



(a) 100-year bins, plain.

(b) 100-year bins, lemmatized.

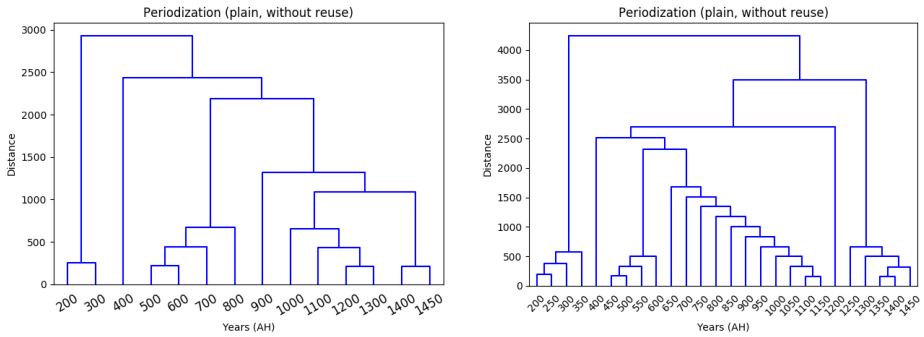
(c) 50-year bins, plain.

(d) 50-year bins, lemmatized.

Fig. 3 Periodization on 50-year and 100-year bins, using plain and lemmatized text. The y-axis shows the cumulative distance between merged clusters.

texts leads again to three large time periods, more or less corresponding to the results with 100-year bins. The periodization based on lemmatized texts exhibits a very large middle period, with comparatively shorter early and late periods. The reason may be that some of the differences between time periods are not as strong when abstracting over the word forms and working with lemmas. Interestingly, the algorithm does not always show the split between texts before and after 1300 AH (1882 CE) that should represent the CA and MSA boundary as it is normally portrayed in the literature. However, all but Figure 3c seem to point towards a differentiation between a modern and pre-modern period.

The algorithm should in principle abstract over genre effects by comparing only shared word embeddings between time periods. However, word embeddings are trained based on their contexts which in turn are influenced by genre. Therefore it should be noted that a consistent grouping which splits the earliest bin from later bins could be due in part to genre effects. The 200 year bin includes all texts from the 1st century, which is dominated by poetic collections. Texts which have the title "diwan", meaning a poetic collection (some poetic collections may have other titles), are primarily found in the first several centuries. Of the 148 distinct works in the first two centuries, 67 or 44% are diwans, but by the year 300, only 14%



(a) 100-year bins, "hollowed" (b) 50-year bins, "hollowed"

Fig. 4 Periodization on 50-year and 100-year bins after removing reused text.

of distinct works are diwans and only 3.4% of all books in the corpus are diwans. Most of the corpus is prose and may differ significantly from poetry in style and use of words in specific contexts. In interpreting the results of the periodization, we should be cautious due to the interaction between genre and text date in the corpus.

6.2.1 Text Reuse and Periodization

The detection of periods of language change can be affected by quotation and general reuse of text chunks from early periods. The text reuse algorithm detects such texts and so we use it here to remove all cases of reuse. The "un-reused" or "hollowed" corpus is obtained by removing, for each detected match, all later cases of reuse, while keeping the earliest instance. Figure 4 shows the resuts of running WENC on the hollowed corpus in 50-year and 100-year bins (showing the plain text version). The results are fairly similar to the periodization results on the full corpus (Figure 3), but there appears to be a clearer separation around 800-900AH, especially in the 100-year bins case. Thus, after removing instances of text reuse, we find a clearer division into different pre-modern time periods.

The separation of the modern period from the pre-modern period becomes more evident once textual quotations are removed, with the 1400s CE (1979 CE to present) clearly separated, and in Figure 4b, the more expected differentiation of 1250 AH (1770 CE) separating from the previous period. This is right at the beginning of the period in which MSA developed. With 100-year bins, 1400 AH would represent authors who died between 1300 AH (1882 CE) and 1979 CE, i.e. a group born around the 1820s to the early 20th century. These represent the second or third generation of modernizing authors. That this would become somewhat more evident in the "hollowed" corpus is logical, as the texts in the corpus are inherently conservative and religious in nature, quoting heavily from earlier works. Removing those quotations makes the modern sections of these texts more salient to the algorithm.

Overall the automatic periodization separates out the earliest and latest texts (200-300 and 1400-1450, sometimes including 1350) from a core bin that occupies

the periods roughly between 400 AH (1009 CE) and 1300 AH (1882 CE). The first bin appears to correspond to Fischer's PSCA era, containing a large quantity of pre-Islamic and early Islamic poetic texts, while the core bin is CA proper. MSA comes surprisingly late, even assuming a 70-year author lifespan. There is some support in the periodizations for the claimed change in the language due to the end of the Abbasid empire in 1258 CE (656 AH), with a cluster break clearest in the "hollowed" periodizations. A surprising result which requires further research is the consistent break around 900 AH (1494 CE). Though the literature has not considered this as a transition period, it does corresponds to the end of the Islamic state in the Iberian peninsula (1492 CE), and the rise of the Ottoman empire (conquest of Syria and Egypt during 1510s CE).

7 Expert Periodization of Arabic

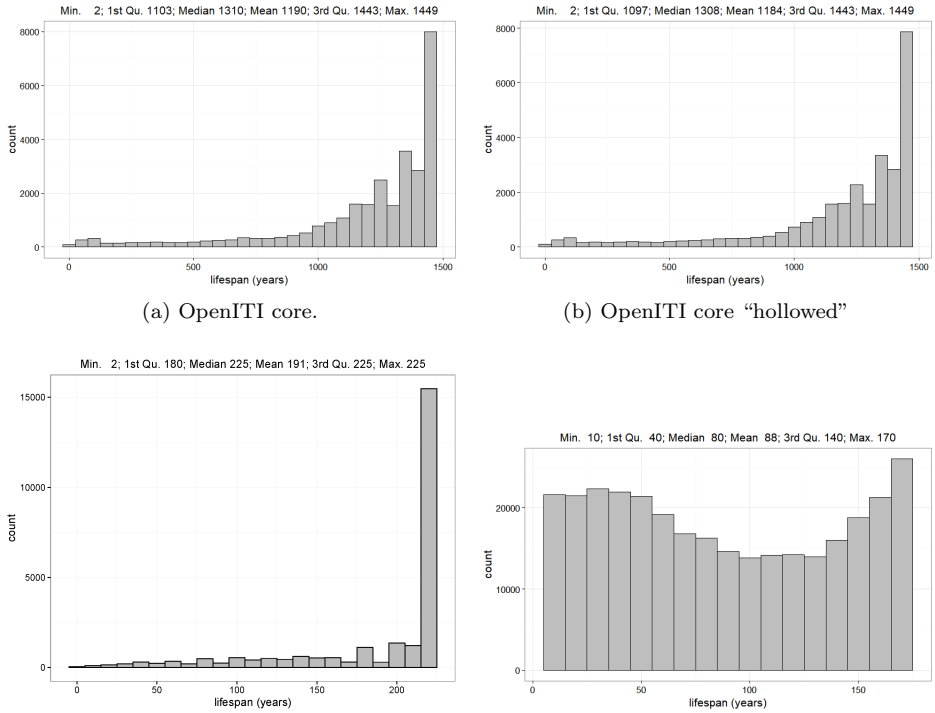
The corpus also supports less automated approaches to periodization that still rely on computational tools. These approaches are particularly useful for assessing the validity of periodizations based on impressionistic analyses of Arabic from earlier publications.

7.1 The Lifespan of Arabic Words

It is possible that the impression of SA as unchanging is actually a myth rather than reality, that SA actually changes just as quickly as other written languages. In order to investigate this question we use the corpus to check whether there is a quantitative difference in the development of Arabic writing and other languages for which historical corpora are available.

To do this, we track the "lifespan" of Arabic and English words in two corpora: OpenITI core and the Corpus of Historical American English (COHA) [21]. We used lemmas rather than words in both cases. For OpenITI, we use the lemmatization provided by Farasa, but since Farasa does not reject non-words, we use MADAMIRA [53] to discard lemmas that are not actually Arabic words (incorrectly spelled words, etc.). For every lemma in the corpus, we find its first and last chronological usages. We discard words which occur only once or in a single year, which may be misspellings (often the case in COHA), or which have apparently short lifespans since they occur in the very last year/decade (in COHA) in the corpus.

Figure 5 shows the difference between Arabic and English word lifespans. The comparison reveals that Arabic words tend to have a very long life span in comparison with English words. In OpenITI the average Arabic word lifespan is 1,190 years (see figures for further descriptive statistics), approximately 83% of the time span of the entire corpus. In COHA, the average English word lifespan is 88 years, about 45% of the overall time span of the 190 year corpus. Of course, COHA is a much narrower corpus, only covering 200 years, from 1810-2000s. To control for the 200 year span of COHA versus the much longer span of our corpus, we also ran the same analysis on the set of all thirteen 200 year spans of the corpus (in 100-year increments, with one 225 years span for the final bucket). All lifespan analyses exhibited a similar trend, with long left-tailed distributions; Figure 5c shows an



(a) OpenITI core.

(b) OpenITI core "hollowed"

(c) OpenITI core 1225AH-1450AH (1810CE-Present)

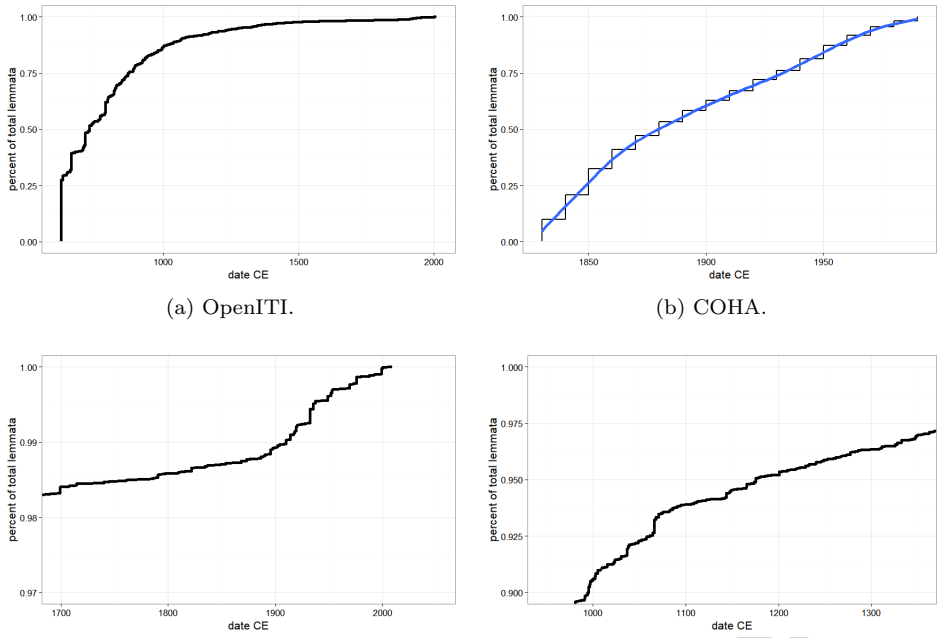
(d) COHA.

Fig. 5 Distribution of word lifespans in Arabic, using (a) OpenITI core, (b) a hollowed version without text reuse, (c) the last 225 years, and (d) the COHA English corpus.

example for the last time span. Morover, under this analysis, the mean lifespan of words was 161 years, nearly double COHA's average word lifespan. These results confirm that Arabic words tend to have longer lifespans than English ones.

SA words do not disappear as quickly from the lexicon as English words do, suggesting that indeed SA does change less quickly than English. This provides quantitative confirmation of an earlier qualitative observation: while Arabic-speaking schoolchildren can read a text from 750 CE, an English-speaking middle school student would have a much more difficult time reading Beowulf, an Old English poem from ca. 1000 CE. It also means that any measures of change in SA will need to be more sensitive than measures used for English in order to produce a meaningful result.

It is possible that the apparently long lifespan of these words is due to extensive quotation of texts including archaic words. In a sense, this does not really matter from the perspective of a language user, since they still must be able to understand the quotation regardless (footnotes may be provided to help with this in some texts, but not all.) To determine the extent that quotation influences word lifespan in Arabic, we ran the same lifespan measurements on the "hollowed" version of the lemmatized corpus with reuse removed. The results are shown in Figure 5b, and



(a) OpenITI. (b) COHA.

(c) OpenITI, modern period. (d) OpenITI, medieval period.

Fig. 6 New lemmas as a percentage of total lemmas over time in Arabic in OpenITI and English in COHA.

differ very little from the results on the normal lemmatized corpus. This shows that the long lifespans of Arabic words are clearly not the result of quotation alone.

This analysis should not be understood to say that as a language (in all of its varieties), Arabic somehow changes more slowly than English. The purpose of this comparison is to contextualize how slowly SA changes over time, and give a sense of the scale of the question of how to periodize Arabic. The availability of COHA, with full lemmatization, means that this was the obvious choice for comparison. In terms of communicative function, SA is extremely different from English, having been a learned language of educated writing with strong social norms against inclusion of non-standard variants, whereas English is far more permissive of non-standard language that reflects actual spoken usage.

7.2 New Words Over Time

Another claim in the literature is that the number of new words increased significantly during the development of MSA, as new terms were needed to refer to European technology and ideas suddenly becoming available in the Arab world due to colonization and modernization efforts [50]. We can use the corpus to confirm that this is indeed accurate, and to investigate whether there are other periods when an increase occurred in the number of new vocabulary items.

Figure 6a shows that, as expected, new lemmas in Arabic developed very rapidly, a corollary of the long lifespan of Arabic words. Compare this to the English results which show a steadier rise (Figure 6b). Zooming into the modern era in Figure 6c, we see that fully 1% of Arabic words were added just in the 20th century (really earlier, since these are date-of-death measures), a much more rapid increase than in the previous two centuries, which added slightly more than 0.5% of lemmas to the lexicon. There appears to be another period of rapid growth at the beginning of the second millennium, with a large jump in vocabulary between 1000 and 1100, and again a jump starting in 1150, but they are not as clear as the increase in the modern period (Figure 6d). Some caution is needed as a single author or work could easily cause these jumps in the vocabulary, but this is an interesting result: the account of Arabic history that claims a decline in the language following the end of the Abbasid Empire empire would predict that vocabulary addition would level at this point, rather than increasing. On the other hand, the breakdown of central authority might decrease standardization of vocabulary and increase diversity in specialized terminology.

7.3 Verifying Previous Periodizations

Pre-Standardized Classical Arabic (PSCA) does have distinguishing linguistic features, though these are largely found in the Quran or in rare poetic attestations. Moreover, most of these are slight differences in assimilation or vocalization, and would not show up in the written text in an easily distinguishable way. One of the few testable claims is that prior to the 8th century CE, the formation of abstract conceptual nouns was done via a phrase, *ElY jhp Al-* 'from the perspective of', but was replaced with a suffix *-yp* '-ity, -ness' in the period of Standardized Classical Arabic (SCA) [5]. Two suggested phrases based on the *ElY jhp Al-* 'from the perspective of' structure are *ElY jhp Al-xyr* 'charity, goodness' and *ElY jhp Al-Edl* 'justice, fairness'. A concordance search on the lemmatized corpus finds these structures are basically unattested in the data with a total of less than 30 attestations, all of which post-date the 8th century CE. Nor is this structure attested in Arabic papyri via the Arabic Papyrology Database.[30] It is therefore unclear where this claim originates since it is so poorly supported by the data.[31]

More testable features are claimed for the break between SCA and PCA. These include the use of the adverb *AyDA* 'also'[32], and the development of adjectives that show a suffix *-Any* such as *jsmAny* 'bodily' and *rwHAny* 'spiritual' [25]. These are relatively insignificant changes, but at least they can now be verified or disproved using OpenITI.

Using the segmentation produced by Farasa to remove extraneous clitics (Section 4.2), we are able to investigate whether these claims are accurate. Figure 7 shows the relative frequencies for the word *AyDA*, and the combined frequencies for the words *jsmAny* and *rwHAny*, as well as several other words suggested in

[30] `http://www.apd.gwi.uni-muenchen.de:8080/apd/project.jsp`.

[31] The source of this reference, [5], cites a work that happens not to be in his bibliography and is therefore difficult to locate.

[32] [25] also suggests the PCA adverb *xASp* 'especially', e.g. *xāṣṣat-an* but this is homographic with the word *xāṣṣa* 'the elite' and so cannot be easily distinguished from one another using computational methods.

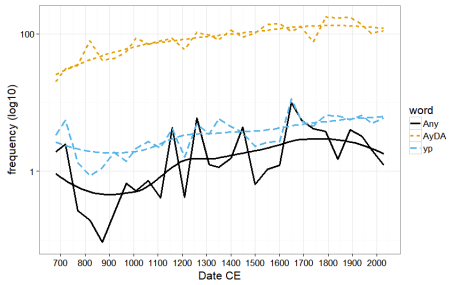


Fig. 7 Relative frequencies of words and suffixes, with actual frequencies and LOESS smoothed lines.

the literature, which use an abstracting suffix *-yp*.[33] For the word *AyDA*, there is no relationship to different eras, with this word functioning as an adverb even in very early texts. In OpenITI our earliest clear attestation is in a text with author DOD of 68 HA (688 CE)[34] It increases in usage over time, but shows no periods of significantly greater growth. It is worth noting that its use was judged negatively, with Ibn al-Sikkit's (d. 858 CE) dictionary providing precise rejoinders to mock anyone who uses the word in its adverbial meaning as 'also' [41]. This establishes that the 'also' meaning is attested at least by the 9th century, but also that this term was undergoing some form of change for it to be subject to meta-linguistic judgment. However, this significantly predates the claimed PCA development date, nor is there a discernible increase in its frequency at that time.

The other two variables do correspond roughly to the claimed PCA era, but increases in their usage actually seem to precede the Mongol conquests in 1258 CE — since all figures here are dates of death, clearly use of these suffixes was on the rise well before that time, and recent research suggests that the Mongols simply finished off a process of decline already underway in Iraq and Iran during that era [58]. The two variables do show remarkable similarity in their frequencies over time, suggesting that the increase in their usage does reflect a change in linguistic eras, even if it begins prior to 1258 CE.

8 Conclusions

The Arabic language has a long and diverse history spanning more than 1400 years. The written Arabic language, Standard Arabic (SA), is often thought to be a more or less monolithic language with little change before the modern period. In this work, we process OpenITI, a large-scale diachronic corpus of Arabic, and investigate the question of periodization in different ways. We identify instances of text reuse in the corpus, develop an automatic periodization algorithm, and investigate existing claims about the periodization of Arabic. We find that although

[33] The words are *mEqwlyp* 'intelligibility',*Eqlyp* 'mentality', *nHwyp* 'specificity', *xyryp* 'charitability', and *Edlyp* 'justice' all suggested by [5].

[34] OpenITI document `0068AbdAllahIbnCabbas.GharibQuran.Shamela0023622`. We find an even earlier attestation in the Arabic Papyrology database, being used as 'also' in papyrus "P.StoetzerSteuerquittungen 2" from 677 CE.

words do persist relatively longer in Arabic than in English, there is evidence for several distinct periods in the language's development.

OpenITI represents the largest publicly available diachronic corpus of Arabic to date. We preprocessed the corpus to make it more amenable to natural language processing, and the results of this are also available for use. The nature of the corpus is such that texts frequently quote from one another, and using an efficient algorithm we were able to identify 292 M words of reused text, nearly 20% of the total corpus. We were able to produce a "hollowed" version of the corpus with this data removed, both for the plain-text and preprocessed versions of the corpus.

This corpus and the tools we develop allow us to answer open questions about the history of Arabic. The corpus allows us to establish that Arabic vocabulary does indeed change more slowly over time than in English. The automatic periodization algorithm we develop confirms established periodizations of Arabic, while suggesting new ones. It shows that the oldest periods of Arabic and the most modern ones are both separate from a core period stretching from approximately 400 CE (1009 CE) until 1300 AH (1882 CE), reflecting the prototypical Classical Arabic (CA). The data from the automatic periodization and from the evidence of new words in Arabic both strongly support established periodizations that divide Modern Standard Arabic (MSA) from CA. Both automatic periodization and computational evaluation of established periodizations provide some support for a break between an early and later Classical Arabic around the 1000s CE, which is somewhat earlier than typically believed, and suggests an additional previously unexplored division in the late 15th century CE.

The processed corpus and the code associated with this work are available to the research community. We hope that future work will illuminate other aspects of the history of the Arabic language, as well as utilize the methods proposed in this work for studying other languages.

Acknowledgements

This research was partly supported by the HBKU Qatar Computing Research Institute (QCRI), as part of a collaboration with the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). Y.B. was also supported by the Harvard Mind, Brain, Behavior Initiative. This research was also partly supported by the Israel Science Foundation (Grant No. 977/16), and by DICTA: The Israel Center For Text Analysis.

References

1. Abdelali, A., Darwish, K., Durrani, N., Mubarak, H.: Farasa: A Fast and Furious Segmenter for Arabic. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 11–16. Association for Computational Linguistics (2016). DOI 10.18653/v1/N16-3003. URL http://aclanthology.coli.uni-saarland.de/pdf/N/N16/N16-3003.pdf
2. Al-Jallad, A.: An outline of the grammar of the Safaitic inscriptions. No. 80 in Studies in Semitic languages and linguistics. Brill (2015)
3. Al-Sulaiti, L.: Designing and Developing a Corpus of Contemporary Arabic. Master's thesis, The University of Leeds, Leeds, UK (2004)
4. Al-Thubaity, A.O.: A 700M+ Arabic Corpus: KACST Arabic Corpus Design and Construction. Lang. Resour. Eval. **49**(3), 721–751 (2015)
5. Ali, A.S.M.: A linguistic study of the development of scientific vocabulary in Standard Arabic. Keegan Paul International (1987)
6. Alrabiah, M., Al-Salman, A., Atwell, E.: The design and construction of the 50 million words KSUCCA. In: Proceedings of WACL2 Second Workshop on Arabic Corpus Linguistics, pp. 5–8 (2013)
7. Arts, T., Belinkov, Y., Habash, N., Kilgarriff, A., Suchomel, V.: arTenTen: Arabic Corpus and Word Sketches. Journal of King Saud University - Computer and Information Sciences **26**(4), 357 – 371 (2014). Special Issue on Arabic NLP
8. Basile, C., Benedetto, D., Caglioti, G., Degli Esposti, M.: A Plagiarism Detection Procedure in Three Steps: Selection, Matches and Squares. In: Stein et al. [69], pp. 19–23. Http://ceur-ws.org/Vol-502
9. Belinkov, Y., Magidow, A., Romanov, M., Shmidman, A., Koppel, M.: Shamela: A large-scale historical arabic corpus. In: Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH at Coling), pp. 45–53. The COLING 2016 Organizing Committee, Osaka, Japan (2016)
10. Bensalem, I., Boukhalfa, I., Rosso, P., Lahsen, A., Darwish, K., Chikhi, S.: Overview of the AraPlagDet PAN@ FIRE2015 Shared Task on Arabic Plagiarism Detection. In: Notebook Papers of FIRE 2015 (CEUR-WS vol. 1587), pp. 111–122. Gandhinagar, India (2015)
11. Björkelund, A., Çetinoğlu, Ö., Farkas, R., Mueller, T., Seeker, W.: (Re) Ranking Meets Morphosyntax: State-of-the-Art Results from the SPMRL 2013 Shared Task pp. 135–145 (2013)
12. Braschler, M., Harman, D. (eds.): Notebook Papers of CLEF 2010 LABs and Workshops. Padua, Italy (2010)
13. Chambers, N.: Labeling Documents with Timestamps: Learning from their Time Expressions. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 98–106. Jeju Island, Korea (2012)
14. Claridge, C.: Historical corpora. Corpus linguistics: An international handbook **1**, 242–259 (2008)
15. Clough, P., Gaizauskas, R.: Corpora and Text Re-Use. In: A. Lüdeling, M. Kytö, T. McEnery (eds.) Handbook of Corpus Linguistics, Handbooks of Linguistics and Communication Science, pp. 1249—1271. Mouton de Gruyter (2009)
16. Clough, P., Gaizauskas, R., Piao, S., Wilks, Y.: Measuring Text Reuse. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), pp. 152–159. Association for Computational Linguistics, Philadelphia, PA (2002)
17. Dalli, A., Wilks, Y.: Automatic Dating of Documents and Temporal Text Classification. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 17–22. Sydney, Australia (2006)
18. Darwish, K., Abdelali, A., Mubarak, H.: Using Stem-Templates to Improve Arabic POS and Gender/Number Tagging. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), LREC 2014, pp. 2926–2931. European Language Resources Association (ELRA) (2014). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/335_Paper.pdf
19. Darwish, K., Mubarak, H.: Farasa: A New Fast and Accurate Arabic Word Segmenter. In: LREC (2016)
20. Darwish, K., Mubarak, H., Abdelali, A., Eldesouki, M.: Arabic POS Tagging: Don't Abandon Feature Engineering Just Yet. In: Proceedings of the Third Arabic Natural Language Processing Workshop, pp. 130–137. Association for Computational Linguistics (2017). URL http://aclweb.org/anthology/W17-1316

21. Davies, M.: The Corpus of Historical American English: 400 million words, 1810-2009. http://corpus.byu.edu/coha (2010)
22. Dubossarsky, H., Tsvetkov, Y., Dyer, C., Grossman, E.: A bottom up approach to category mapping and meaning change. In: V. Pirrelli, C. Marzi, M. Ferro (eds.) Word Structure and Word Usage. Proceedings of the NetWordS Final Conference (2015)
23. Elewa, A.H.: Collocation and Synonymy in Classical Arabic: A Corpus-based Approach. Ph.D. thesis, The University of Manchester, Manchester, UK (2004)
24. Ferrando, I.: History of Arabic. In: K. Versteegh (ed.) Encyclopedia of Arabic Language and Linguistics, vol. 2, pp. 604–611. Brill, Leiden (2007)
25. Fischer, W.: Classical Arabic. In: K. Versteegh (ed.) Encyclopedia of Arabic Language and Lingusitics, vol. 1, pp. 397–405. Brill, Leiden (2006)
26. Gries, S.T., Hilpert, M.: Variability-based Neighbor Clustering: A bottom-up approach to periodization in historical linguistics. In: T. Nevalainen, E.C. Traugott (eds.) The Oxford Handbook of the History of English, pp. 134–144. Oxford University Press, Oxford (2012)
27. Grozea, C., Popescu, M.: The Encoplot Similarity Measure for Automatic Detection of Plagiarism - Notebook for PAN at CLEF 2011. In: V. Petras, P. Forner, P. Clough (eds.) Notebook Papers of CLEF 2011 LABs and Workshops. Amsterdam, The Netherlands (2011)
28. Habash, N., Rambow, O.: Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In: Proceedings of ACL (2005)
29. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In: Proceedings of the Second International Conference on Arabic Language Resources and Tools (2009)
30. Hall, D.L.W., Durrett, G., Klein, D.: Less Grammar, More Features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 228–237 (2014)
31. Hamilton, W.L., Leskovec, J., Jurafsky, D.: Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1489–1501. Association for Computational Linguistics (2016). DOI 10.18653/v1/P16-1141. URL http://aclanthology.coli.uni-saarland.de/pdf/P/P16/P16-1141.pdf
32. Hammo, B., Yagi, S., Ismail, O., AbuShariah, M.: Exploring and exploiting a historical corpus for Arabic. Language Resources and Evaluation **50**(4), 839–861 (2016). DOI 10.1007/s10579-015-9304-9. URL https://doi.org/10.1007/s10579-015-9304-9
33. Holes, C.: Modern Arabic: Structures, Functions, and Varieties. Georgetown University Press, Washington, D.C. (2004)
34. Ji, M.: A corpus-based study of lexical periodization in historical Chinese. Literary and Linguistic Computing **25**(2), 199–213 (2010)
35. Joachims, T.: Training Linear SVMs in Linear Time. KDD '06, pp. 217–226. ACM, New York, NY (2006)
36. de Jong, F., Rode, H., Hiemstra, D.: Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences (2005)
37. Kasprzak, J., Brandejs, M.: Improving the Reliability of the Plagiarism Detection System. Lab Report for PAN at CLEF 2010. In: Braschler and Harman [12]
38. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The Sketch Engine. In: Proceedings of EURALEX (2004)
39. Kim, Y., Chiu, Y.I., Hanaki, K., Hegde, D., Petrov, S.: Temporal Analysis of Language through Neural Language Models. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, pp. 61–65. Association for Computational Linguistics, Baltimore, MD, USA (2014). URL http://www.aclweb.org/anthology/W14-2517
40. Kulkarni, V., Al-Rfou, R., Perozzi, B., Skiena, S.: Statistically Significant Detection of Linguistic Change. In: Proceedings of the 24th International World Wide Web Conference, WWW '15 (2015)
41. Lane, E.W.: Arabic-English Lexicon. Willams & Norgate (1863)
42. Lentin, J.: Middle Arabic. In: K. Versteegh (ed.) Encyclopedia of Arabic Language and Linguistics, vol. 1, online edition edn., pp. 87–96. Brill, Leiden (2006)
43. Li, W.P.: Language Technologies for Understanding Law, Politics, and Public Policy. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2016)

44. Magidow, A.: A Digital Philological Investigation of the History of *hā hunā* Constructions. Romano-Arabica **16**, 239–256 (2016)
45. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR **abs/1301.3781** (2013). URL `http://arxiv.org/abs/1301.3781`
46. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. CoRR **abs/1310.4546** (2013). URL `http://arxiv.org/abs/1310.4546`
47. Mikolov, T., tau Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013). Association for Computational Linguistics (2013). URL `http://research.microsoft.com/apps/pubs/default.aspx?id=189726`
48. Mikolov, T., Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. In: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, pp. 746–751 (2013). URL `http://aclweb.org/anthology/N/N13/N13-1090.pdf`
49. Muhr, M., Kern, R., Zechner, M., Granitzer, M.: External and Intrinsic Plagiarism Detection using a Cross-Lingual Retrieval and Segmentation System. In: Braschler and Harman [12]
50. Newman, D.L.: The Arabic Literary Language: the nahḍa and beyond. In: J. Owens (ed.) The Oxford Handbook of Arabic Linguistics, pp. 472–494. Oxford University Press, Oxford (2013)
51. Niculae, V., Zampieri, M., Dinu, L., Ciobanu, A.M.: Temporal Text Ranking and Automatic Dating of Texts. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers, pp. 17–21. Gothenburg, Sweden (2014). URL `http://www.aclweb.org/anthology/E14-4004`
52. Osama Hamed, T.Z.: A Survey and Comparative Study of Arabic Diacritization Tools. JLCL **32**(1), 27–47 (2017)
53. Pasha, A., Al-Badrashiny, M., Diab, M., Kholy, A.E., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.: MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp. 1094–1101. Reykjavik, Iceland (2014). URL `http://www.lrec-conf.org/proceedings/lrec2014/pdf/593_Paper.pdf`
54. Popescu, O., Strapparava, C.: SemEval 2015, Task 7: Diachronic Text Evaluation. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 870–878. Denver, Colorado (2015). URL `http://www.aclweb.org/anthology/S15-2147`
55. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An Evaluation Framework for Plagiarism Detection. In: C.R. Huang, D. Jurafsky (eds.) Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), pp. 997–1005. COLING 2010 Organizing Committee, Beijing, China (2010)
56. Rashwan, M.A., Al-Badrashiny, M.A., Attia, M., Abdou, S.M., Rafea, A.: A Stochastic Arabic Diacritizer Based on a Hybrid of Factorized and Unfactorized Textual Features. Trans. Audio, Speech and Lang. Proc. **19**(1), 166–175 (2011). DOI 10.1109/TASL.2010.2045240. URL `http://dx.doi.org/10.1109/TASL.2010.2045240`
57. Rehurek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, Malta (2010)
58. Romanov, M.: Algorithmic Analysis of Medieval Arabic Biographical Collections. Speculum **92**(S1), S226–S246 (2017)
59. Romanov, M., Miller, M.T., Savant, S.B.: OpenITI – machine-actionable scholarly corpus of premodern Islamicate texts (2017–ongoing). URL `https://openiti.github.io`
60. Romanov, M.G.: Computational Reading of Arabic Biographical Collections with Special Reference to Preaching in the Sunni World (661–1300 CE). Ph.D. thesis, University of Michigan, Ann Arbor, MI, USA (2013)
61. Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., Moschitti, A.: Language processing and learning models for community question answering in arabic. Information Processing & Management (2017). DOI https://doi.org/10.1016/j.ipm.2017.07.003

62. Sagi, E., Kaufmann, S., Clark, B.: Semantic Density Analysis : Comparing word meaning across time and phonetic space. In: Proceedings of the EACL 2009 Workshop on GEMS: Geometrical Models of Natural Language Semantics, pp. 104–111 (2009)

63. Scherbinin, V., Butakov, S.: Using Microsoft SQL Server Platform for Plagiarism Detection. In: Stein et al. [69], pp. 36–37. Http://ceur-ws.org/Vol-502

64. Schneider, N., Mohit, B., Oflazer, K., Smith, N.A.: Coarse Lexical Semantic Annotation with Supersenses: An Arabic Case Study. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2, ACL '12, pp. 253–258. Association for Computational Linguistics, Stroudsburg, PA (2012). URL http://dl.acm.org/citation.cfm?id=2390665.2390726

65. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966). DOI 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451

66. Shmidman, A., Koppel, M., Porat, E.: Identification of Parallel Passages Across a Large Hebrew/Aramaic Corpus. arXiv preprint arXiv:1602.08715 (2016)

67. Shoufan, A., Alameri, S.: Natural Language Processing for Dialectical Arabic: A Survey. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 36–48. Beijing, China (2015). URL http://www.aclweb.org/anthology/W15-3205

68. Smith, D.A., Cordell, R., Dillon, E.M., Stramp, N., Wilkerson, J.: Detecting and Modeling Local Text Reuse. In: Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '14, pp. 183–192. London, United Kingdom (2014). URL http://dl.acm.org/citation.cfm?id=2740769.2740800

69. Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.): SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009), vol. 502. CEUR-WS.org, San Sebastian, Spain (2009). Http://ceur-ws.org/Vol-502

70. Wijaya, D.T., Yeniterzi, R.: Understanding semantic change of words over centuries. Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web - DETECT '11 p. 35 (2011). DOI 10.1145/2064448.2064475. URL http://dl.acm.org/citation.cfm?doid=2064448.2064475

71. Wilkerson, J., Smith, D., Stramp, N.: Tracing the Flow of Policy Ideas in Legislatures: A Text Reuse Approach. American Journal of Political Science **59**(4), 943–956 (2015). DOI 10.1111/ajps.12175. URL http://dx.doi.org/10.1111/ajps.12175

72. Zack, E., Schippers, A. (eds.): Middle Arabic and Mixed Arabic: Diachrony and Synchrony. Brill Academic Publishers, Leiden (2012)

73. Zaghouani, W.: Critical Survey of the Freely Available Arabic Corpora. In: Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools (2014)

74. Zemánek, P., Milička, J.: Ranking Search Results for Arabic Diachronic Corpora. Google-like search engine for (non)linguists. In: Proceedings of CITALA 2014 (5th International Conference on Arabic Language Processing). Association for Computational Linguistics (2014)

75. Zemánek, P., Milička, J.: Quotations, Relevance and Time Depth: Medieval Arabic Literature in Grids and Networks. In: Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL), pp. 17–24. Gothenburg, Sweden (2014). URL http://www.aclweb.org/anthology/W14-0903

76. Zerrouki, T., Balla, A.: Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. Data in Brief **11**, 147 – 151 (2017). DOI https://doi.org/10.1016/j.dib.2017.01.011. URL http://www.sciencedirect.com/science/article/pii/S2352340917300112

A Mathematical Notation

Symbol	Meaning
K / M / G	10^3 / 10^6 / 10^9
$\mathcal{T}$	Collection of texts
$\mathcal{P}$	List of texts organized in time periods
P_i	Texts belonging to time period i
W_1, W_2	Word embedding matrices
Q, U, V, Σ	Matrices
I	Identity matrix
Q^T	Transpose of Q
$\|\|W\|\|$	Matrix norm
$\|\|W\|\|_F$	Frobenius norm

B Arabic examples

Boiler Plate Matches The following examples illustrate the boilerplate passages identified in the first step of the text reuse algorithm (Section 5). The first is a chain of transmission that occurs 2,747 times in the corpus. The second is a part of a Hadith that occurs 221 times.

Hdvny mHmd bn Emrw qAl vnA >bw EASm qAl vnA EysY wHdvny AlHArv qAl vnA AlHsn qAl vnA wrqA' jmyEA En Abn >by njyH En mjAhd ...

[1] Muhammad b. 'Amr reported to me, saying: Abu 'Asim reported to us, saying: 'Isa reported to me [2] al-Harith reported to me, saying: al-Hasan transmitted to us, saying: [1&2] Warqa' transmitted to all of us on the authority of Ibn Abi Najih, on the authority of Mujahid..."'"

mn sy}At >EmAlnA mn yhdh Allh flA mDl lh wmn yDll flA hAdy lh w>$hd >n lA <lh <lA Allh wHdh lA $ryk lh w>$hd >n mHmdA Ebdh wrswlh

[I seek refuge in god] from the evil of our deeds; he who God guides rightly cannot go astray; he who goes astray cannot be led aright; I witness that there is no God but God alone with no partner, and that Muhammad is His servant and His Prophet

Near-matches The next example is of a near-match with several differences. The first is the original, from `0292Yacqubi.TarikhYacqubi.Shia003468Vols-ara1`:

lys Tlby llElm TmEA fy blwg qASyth wAlAsjylA' ElY gAyth wlkn Altms $y}A lA ysE jhlh wlA yHsn bAlEAql.

My search for knowledge is not greed to reach its utmost, or to seize its aim. Rather, I seek something of which ignorance is widespread, and which the wise dare not contradict.

As identified in a later text (`1371MuhsinCamili.AcyanShica.Shia003636Vols`), with differences italicized:

lys Tlby llElm TmEA fy blwg qASyth wAlAsjylA' ElY *nhAyth* wlkn *mErfp mA* lA ysE jhlh wlA yHsn bAlEAql.

My search for knowledge is not greed to reach its end, or to seize its aim. Rather, knowledge of that of which ignorance is widespread and which the wise dare not contradict.

Glossary

Notation	Description
Abbasid Empire	The second great Islamic dynasty (750-1258) that replaced the Umayyads (661-750); the formation and early development of the written Arabic tradition takes place during the Abbasid period of Islamic history.
AH	Anno Hegirae. Also known as Islamic Hijri, it is a lunar calendar. 1439AH corresponds to the period from September 2017 to September 2018 AD
boilerplate	A text fragment that is used in multiple contexts with minimal changes from the original
CA	Classical Arabic; the language used in pre-modern texts
CE	Common Era
COHA	Corpus of Historical American English
diacritization	Restoration of diacritics, short vowels and other symbols commonly omitted when writing Arabic
diwan	Here, a collection of poetry
exegesis	Interpretation of scripture
Hadith	Reports about sayings and deeds of the Prophet Muhammad and his companions
JK	Al-Jami‘ Al-Kabir, a digital library of Arabic texts
LOESS	A non-parametric regression method.
matres lectionis	Consonants used in some Semitic languages to indicate a vowel
MSA	Modern Standard Arabic
OpenITI	Open Islamicate Texts Initiative, https://openiti.github.io/
OpenITI core	OpenITI corpus with one file per work
OpenITI full	OpenITI corpus with all files including multiple editions/versions of the same works
PCA	Post-Classical Arabic
PSCA	Pre-Standardized Classical Arabic
PATB	Penn Arabic Treebank
SA	Standard Arabic
Shamela	Al-Maktaba Al-Shamela (“The Complete Library”), a digital library of Arabic texts
SCA	Standardized Classical Arabic
SVD	Singular value decomposition
SVM	Support vector machine
Syriac	A dialect of Middle Aramaic that appeared in the early 1st century CE
VNC	Variability-based Neighbor Clustering
WENC	Word-Embedding-based Neighbor Clustering