Arabic Data Science Toolkit: An API for Arabic Language Feature Extraction

Paul Rodrigues, Valerie Novak, C. Anton Rytting, Julie Yelle, Jennifer Boutz
University of Maryland
Center for Advanced Study of Language
College Park, MD
{prr,crytting,jyelle,jboutz}@umd.edu

Abstract

We introduce Arabic Data Science Toolkit (ADST), a framework for Arabic language feature extraction, designed for data scientists that may not be familiar with Arabic or natural language processing. The functions in the toolkit allow data scientists to extend their algorithms beyond lexical or statistical methods and leverage Arabic-specific linguistic and stylistic features to enhance their systems and enable them to reach performance levels they might receive on languages with more resources, or languages with which they have more familiarity.

Keywords: Arabic, Data Science, Social Media

1 Introduction

Data scientists working on language processing problems prefer statistical methods, because they are language-independent. Best performing methods in document classification problems, however, utilize features that require knowledge of the language text being analyzed. This is especially true with Arabic. Statistical methods, such as bag of words, often are not sufficient for Arabic due to the language's morphological complexity (Kulick et al., 2006). (One example of morphological knowledge benefiting statistical parsing may be found in Yuval et al. (2013).) Since data scientists can already implement linguistic-universal statistical methods with existing software [1], this toolkit focuses on those features that require linguistic or psychological knowledge and the resources to drive them. The Arabic Data Science Toolkit (ADST) is developed primarily in Python, using Flask [2] to run as a web service server. The result is an API accessible via HTTP, allowing access that is not restricted by programming language. Inputs and outputs to the methods are transmitted in a JSON-like format. While intended to run as a server, individual features are accessible via individual scripts implemented in Python and Perl.

Data scientists may throw a large number of features at a classification problem, without initial regards to efficiency, and use feature selection methods to determine which of these features provide the most signal to their problems. They can then choose a subset of those features that are the most useful for a production system, in order to speed up execution time in a repetitive or realtime/online use case. We designed ADST for the flexibility to extract all features, feature sets, or specific individual features, as required.

Language-specific resources are somewhat difficult to find for those unfamiliar with a given language or with natural language processing (NLP) in general. Toolkits such as the Natural Language Toolkit (NLTK)[3] make this easier by uniting disparate tools under one framework. Toolkits designed for multiple languages, such as NLTK, tend to be popular, due to both ease of discovery, and language-flexibility. These toolkits may link to tools with language-specific models, but they tend to prefer linking to tools with open-source licensing. Due to licensing restrictions, they may not always integrate the highest performing Arabic tools (e.g. MADAMIRA). ADST is interested in integrating with the best performing Arabic NLP tools.

A similar tool developed for English is *Structured Programming for Linguistic Cue Extraction* (SPLICE) (Moffit and Giboney, 2015). This tool is a web-hosted API and GUI for linguistic feature extraction. The tool includes Parts of Speech (e.g. Number of Nouns, Verb Ratio), Immediacy (e.g. Number of Passive Verbs, Passive Verb Ratio), Tense (e.g. Past Tense Count, Past Tense Ratio), Positive and Negative Self Evaluation Lexicons, Deference Lexicons (e.g. Asks Permission variable, Submissiveness Ratio), affect lexicons, spoken word counts (e.g. hedging verb count, disfluency count), sentiment lexicon counts, and calculates scores for common English readability metrics (e.g. Flesch-Kincaid Grade Level).

From the beginning, we felt it would be useful to target informal social media language, which poses some unique challenges for Arabic. Many of the tools designed for Arabic focus on Modern Standard Arabic (MSA), while much of the conversation on social media exhibit colloquial varieties. Since many new Arabic tools have focused on adding Egyptian dialect components, and much of the conversation on social media does include Egyptian dialect, we aimed to build a tool that would work on either MSA or Egyptian. When we

[1] Such as scikit-learn's CountVectorizer `http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`

[2] `http://flask.pocoo.org`

[3] `http://www.nltk.org`

discover a feature that has regional variants which are easy to add, we have elected to include them. While we include colloquial language use, our toolkit does not process Arabic chat alphabet, or Arabic that is written in Latin script.

2 Feature Focus

Document classification utilizes features extracted from text to train a model that can associate text to a label in a class of labels. Many classification problems utilize the same features, but features may be ranked or weighted differently to predict the different classes. We settled on two common document classification problems for our initial target in the ADST, emotion classification and authorship attribution. Below, we describe the two classification problems, and explain their similarities and differences.

2.1 Emotion classification

Emotion classification is a class of problems concerned with the identification of an author's latent emotional state at the time of writing. For example, the presence of هههه *hahahaha* may indicate the author is happy. Labels in the emotion class differ in implementations of emotion classification problems due to a wide assortment of emotion category sets available in the emotion literature. One core set of emotions often used in classification implementations are the Ekman emotions (Ekman, 1992) which consist of anger, disgust, fear, happiness, sadness, and surprise.[4]

In addition to category labels, some who work on emotion classification are interested in the intensity of the emotion. You could imagine that somebody who typed هههههههه *hahahahahahaha* may have been more happy than somebody who merely typed هههه *hahaha*. In the intensity use case, not only the presence of a laugh class of text, but the length of that text is important signal.

2.2 Authorship attribution

Authorship attribution compares an author's unique writing characteristics to a set of existing author examples and selects the closest author from that set. In this document classification problem, the mere presence of laughter, or the length of laughter in a text, is not an adequate signal to author attribution. Some authors may exhibit laughter like هههههه *hahahaha*, while some might laugh like هيهيهي *heeheehee*. These distinctions could supply useful information for some classification problems and require the ability to return them explicitly.

3 External Resources

The ADST utilizes several publicly-available Arabic resources to power the features it produces. These resources are high quality, but do not by default prepare feature matrices appropriate for document classification.

MADAMIRA[5] MADAMIRA is a toolkit for Arabic morphological analysis, tokenization, lemmatization, and diacritization (Pasha et al., 2014). What sets MADAMIRA apart from other tools that perform these functions is that it takes word context into account, disambiguating each word in relationship to the sentence it is a member of. MADAMIRA contains models for both MSA and Egyptian. POS Settings are configurable in the MADAMIRA configuration file. At the present time, we are utilizing both the ATB and ATB4MT tokenization schemes and the Egyptian configuration file, but this could be modified in future classification experiments.

NRC Word-Emotion Association Lexicon[6] The NRC (National Research Council) Word-Emotion Association Lexicon is a lexical database of individual words to corresponding emotions and sentiment (Mohammad and Turney, 2013). It labels for each word the emotion categories of joy, sadness, anger, fear, trust, disgust, surprise, and anticipation, as well as for positive and negative sentiment. The database was crowdsourced in English, and then translated into 39 additional languages, automatically, and without context, using Google Translate.

Arabic Word List for Spell Checking For features involving misspellings and creative spellings (such as word elongation) we make use of a freely available Arabic word list (Attia et al., 2014)[7]. This wordlist is described as automatically generated from the AraComLex open-source finite state transducer (Attia et al., 2014), enriched with words from the Arabic Gigaword corpus, and validated against the Microsoft Word spell checker.

4 Feature sets

This section discusses the features included in the ADST, and the research required to create them. We start by introducing features relevant to social media and useful when analyzing informal text. Next, we briefly discuss Emotion category features, which are useful for emotion identification. Arabic stylistic features follow, which help identify individual nuances in writing. A section on word category lexicons is next, which provides some topic identification features to the data scientist. This section also includes a description of the method we use to match dialectal Arabic text to words in lexicons. A section on morphological features is included, whose features can be used for a variety

[4] The Ekman emotions are defined on facial expression differences, and are not defined for text annotation. Additionally, psychologists have proposed many alternative category sets.

[5] https://secure.nouvant.com/columbia/technology/cu14012/license/492

[6] http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[7] Arabic Wordlist 1.6, downloaded from www.attiaspace.com on 2016-11-11, also available from http://sourceforge.net/projects/arabic-wordlist/ Version 1.6 contains 9,196,215 fully inflected words, slightly fewer than the version described in the paper (presumably because of further vetting).

of classification problems. Finally, we provide a section on spelling and typographical features, useful for author identification.

4.1 Social Media Features

4.1.1 Laughter

We developed a set of scripts capable of detecting strings that represent laughter and calculating their length to assess emotion intensity. Just as laughter may be represented more than one way in the written form in English – for example, "hahaha," "heeheehee," or "ho ho ho!" – laughter is represented multiple ways in Arabic. Given that Arabic is rich in uvular, pharyngeal, and glottal phonemes (sounds produced at the back of the throat) and contains a distinction between short and long vowels, the possibilities for representing laughter in the Arabic script are particularly wide-ranging. Our Python scripts were designed to detect as many strings as possible that are likely to represent laughter in Arabic script, based on observations of the type of laughter attested in Arabic-language social media usage. Some representations of laughter in the Arabic script are uniliteral strings (i.e. strings consisting of a single character) formed by the repetition of a single voiceless fricative – either the velar/uvular خ *khā'*, the pharyngeal ح *ḥā'*, or the glottal ء *hā'* – in which the presence of a short vowel is implied. We thus developed a script to detect and count the length of laughter strings formed by repeating three times or more any one of these Arabic consonants, e.g. خخخ *khakhakha*, ححح *ḥaḥaḥa*, and هههه *hahaha*. We also developed numerous scripts designed to work with laughter strings involving more than one character. For multiliteral strings, the set of laughter-associated characters was expanded to include the consonants ق *qāf*, likely representing its colloquial pronunciation as a glottal stop) and ع *'ayn*, as well as the long vowels ا *alif* and ي *yā'*. Our scripts were designed to take into account that an Arabic biliteral laughter string may consist of either a consonant–long vowel pair (e.g. هاها, *hāhā*) or a consonant-consonant pair in which the presence of a short vowel is implied (e.g. معهع, *ha' ha'*). Our scripts were also developed to detect such strings regardless of whether the repetition of and alternation between the two characters is regular (e.g. هخهخه, *hakhahakhaha*) or irregular (e.g. هخهخههه, *hakhahakhahahaha*). We also specifically created a script to detect a biliteral string in which the consonant jīm is regularly alternated with the letter alif (e.g. جاجاجا, *jājājā*) and whose appearance in Spanish-language contexts and adjacency to laughter emoji suggests that it is an Arabic transliteration of Spanish-language laughter (i.e. jaja). Also attested in social media contexts are triliteral, quadriliteral, and quintiliteral strings that combine any of three, four, and five characters, respectively, from the following set of laughter-associated characters: ء *hā'*, ح *ḥā'*, خ *khā'* , ع *'ayn*, and ق *qāf*, and ا *alif*. We are currently adding these multiliteral laughter categories to ADST.

Some tools, like the English syntactic parser TweeboParser (Owoputi et al., 2013)[8] do recognize laughter, but conflate the laughter categories to a single category, which removes the uniqueness of the laughter type signal.

In addition to laughter scripts, we wrote scripts to detect ululation, another type of emotional vocalization that is rooted in Arab culture (as well as the cultures of many other African and Asian societies). Ululation is a high-pitched, elongated vocalization with a trilling quality, is attested on both positive occasions (e.g. celebrating a wedding) and negative ones (e.g. mourning at a funeral). Thus, it can connote either joy or sorrow, depending upon the context in which it is used. Ululation connotes high intensity of emotion regardless of whether the nature of that emotion is positive or negative. We based our ululation-detection scripts on two main written variants attested in Arabic-language Twitter content: يويويو (yūyūyūyū) and لولولي (lūlūlūlī). The latter variant even had a hashtag (#لولولي) associated with it.

4.1.2 Emoji

Emoji are Unicode characters that can be used to add additional context for interpretation of a written text. ADST includes two feature sets around Emoji. The first returns a histogram of the emoji found in a text. The second utilizes a compiled lexicon associating Emoji to the emotions that may be communicated by that Emoji. This relies on a previous study which had English- and Arabic-native speakers annotate for whether emoji in context evoked sadness, contempt/disgust, affection, fear, happiness, positive humor, anger, confidence/pride, or no emotions. The ADST utilizes this lexicon to return a histogram of the emotional categories that may be signaled by an author's emoji usage.

4.2 Emotion Lexicon

In addition to the emoji lexicon, the ADST uses the NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2013) to provide emotion information of the words used in the text. The ADST examines the lexicon and returns word counts for each of the emotive categories they used (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation). Additionally, it returns word counts for the two sentiment categories used in the lexicon (positive, negative).

4.3 Arabic-focused Stylistic Features

Many author attribution systems utilize features that attempt to capture linguistic style. These stylistic features might consist of punctuation frequencies, or word frequencies. We have taken some of the features used in the English author attribution literature, and extended them for Arabic. Additionally, Arabic has several unique stylistic features, that we extract in ADST for future research.

A punctuation script, for example, outputs the type and frequency of each type of punctuation. These

[8] http://www.cs.cmu.edu/~ark/TweetNLP/

include English and Arabic versions of the question mark (? and ؟) as well as punctuation that the languages have in common. A quotation script searches and counts the symbols and punctuation used as quotation markers.

Arabic characters appear in different shapes, depending on their position in a word. The Unicode code pages for Arabic characters include a code page that utilizes this contextual information to provide the shaping of the character, as well as a code page for presentation forms. Arabic in the Presentation Form unicode block can look identical to characters in the contexual Unicode block, but consist of different underlying character codes, which can hinder pattern matching on Arabic. Some older devices utilize presentation mode exclusively, which may provide additional information for the identification of authorship. A presentation form script identifies these alternate Arabic characters by searching in the Unicode range of \uFB50-\uFDFF and \uFE70-\uFEFF for Unicode points for Arabic letters which are not the canonical form (and hence would not normally be produced by typical input methods) but which are visually identical to canonical Unicode Arabic text.

Unicode also has full Arabic words as single Unicode points. For example, محمد *Muḥammad* is a name/word that can be denoted by the single unicode point \ufdf4. The phrasal characters script identifies the type and frequency of these characters in the text.

There are two numbers scripts, one of which outputs a frequency of Arabic numbers and the other a frequency of Hindi numbers used in the text. Lastly, there is a script that counts the frequency of the tatweel (the Arabic lengthening character) and the frequency of Arabic diacritics (short vowels).

These scripts provide Arabic-specific features of text to the data scientist, who might not know that they are salient to the document classification problem.

4.4 Word Category Features

We developed several custom lexicons for this toolkit that target feature sets of interest to emotion identification and authorship attribution. This allows a data scientist to submit their text to determine if words in their text belong to particular pre-defined categories.

To build the lexicons, we drew from various published dictionaries of MSA and the Badawi and Hinds (1986) dictionary of Egyptian Arabic. We used web-based glossaries for topics that were not covered adequately in print dictionaries, in particular insulting and profane language. We also supplemented the lexicons with researcher-identified multi-word entries from social media data. Many of these expressions are not included in print resources.

For each lexicon type, we provide an MSA and Egyptian Colloquial (ECA) lexicon (e.g., MSA Pious Expressions Lexicon and ECA Pious Expressions Lexicon). These lexicons were created independently of each other and the entries do not correspond exactly. The ECA Abusive Language Lexicon encompasses the ECA version of the content of both the MSA Abusive Language and Targeted Abusive Language Lexicon.

We chose to keep discrete categories separate in the lexicons, however, as users of the toolkit may choose to merge lexicons to create different feature combinations depending on their interests. For example, the Polite Expressions, Honorifics, and Pious Expressions Lexicons could be combined to create a single Respectful Language/Politeness Lexicon and feature. Similarly, the various categories of lexicons related to abusive language might be merged into a single lexicon.

4.4.1 Lexicons and Lexicon Expansion

Arabic inflects words for gender and number, and has additional morphological complexity resulting from the affixation and concatenation of conjunctions, prepositions, and the definite article. This complexity makes it difficult to search a text for a word. In order to identify documents containing words and phrases from the word category lexicons, we developed machine-readable versions of these lexicons, expanding the simple lexicons to include Arabic's morphological complexity, as well as commonly occurring orthographic variations and common misspellings.

The application of regular expressions designed to handle orthographic variation was done by searching for instances of specific characters and patterns (e.g., ة *ta marbuta*, ي *yaa'*, and أ *hamza-on-alif*) and replacing them with the appropriate regular expression (respectively, [ةه], [یی], and [آأإا]).

The morphological expansion of Arabic words was handled according to two basic part-of-speech categories (verbs and nouns) and their corresponding morphological behavior. Therefore, two basic approaches were used: one for verbal inflectional paradigms and the other for nominal inflectional paradigms (a category that includes adjectives).

For morphological expansion of verbal forms two basic paradigms were used: the perfective (الماضي) and the imperfective (المضارع). In the following regular expressions, the string STEM represents the verb stem. It should be noted that although these paradigms are based on MSA morphology, they have been extended to cover dialectal Arabic morphology, such as the Egyptian future particle ح and the Levantine and Iraqi progressive aspect particles ب (Levantine) and د (Iraqi), as well as the negative ش enclitic used in Egyptian and some Levantine dialects.

Regular expression for the perfective verb can be found in Line 1 of Figure 1. Regular expressions for the imperfective verb can be found in Line 2 of Figure 1. 1st pers. sg./pl., 2nd pers. masc. sg, and 3rd pers. masc./fem. sg 2nd pers.masc.pl and 2nd pers.fem.sg can be found in Line 3 of Figure 1. 3rd pers.masc.pl can be found in Line 4 of Figure 1.

Note that our lexicon did not include irregular verb paradigms, hence we wrote our regular expressions to cover only regular paradigms.

For morphological expansion of nominal forms two basic paradigms were used: nominal form with posses-

```
1  /^[وف]?STEM(?:ت|تي|تى|وت|ا?و|ا?ن)?(?:ه|ها|هم|هن|ك|كي|كى|كم|كوا|نا|ني|نى)?ش?$/
2  /^[وف]?[حدب]?[اتني]STEM(?:ه|ها|هم|هن|ك|كي|كى|كم|كوا|نا|ني|نى)?ش?$/
3  /^[وف]?[حدب]?تSTEM(?:وا?|ي|ى|ون|ين)(?:ه|ها|هم|هن|ك|كي|كى|كم|كوا|نا|ني|نى)?ش?$/
4  /^[وف]?[حدب]?يSTEM(?:وا?|ون)(?:ه|ها|هم|هن|ك|كي|كى|كم|كوا|نا|ني|نى)?ش?$/
5  /^[وف]?(?:[بلكع]|[بكع]?ال|[بلكع]?ال|لل)?STEM$/
6  /^[وف]?(?:[بلكع]|[بكع]?ال|[بلكع]?ال|لل)?STEM(?:ين|ات)$/
7  /^[وف]?(?:[بلكع]|[بكع]?ال|[بلكع]?ال|لل)?STEM(?:ة|تين|ات)$/
8  /^[وف]?(?:[بلكع]|[بكع])?STEM(?:ه|ها|هم|هن|ك|كم|كوا|نا|ي)$/
9  /^[وف]?(?:[بلكع]|[بكع])?STEM(?:ا?ت(?:ه|ها|هم|هن|ك|كم|كوا|نا|ي))$/
```

Figure 1: Regular expressions for morphological expansion

sive pronoun enclitic, and nominal form without possessive pronoun enclitic. This latter paradigm allows for the presence of the definite article proclitic (which is incompatible with the possessive pronoun enclitic). Both paradigms allow for insertion (between the nominal stem and the possessive pronoun enclitic) of number and gender marker enclitics (i.e. masc. dual, masc. pl., fem. sg., fem. dual, and fem. pl.).

Regular expressions for the nominal form without possessive pronoun enclitic can be found in Line 5 of Figure 1 masc. sg. and masc. dual/pl can be found in Line 6 of Figure 1. fem. sg./dual/pl. can be found in Line 7 of Figure 1. Regular expressions for the nominal form with possessive pronoun enclitic. Note that these paradigms represent Arabic dialect usage. (The MSA masc. forms include the dual and plural, but these are used only in formal contexts). masc. can be found in Line 8 of Figure 1. fem. sg/pl. can be found in Line 9 of Figure 1

Phrasal entries are currently represented with regular expressions that cover only minimal inflection.

This morphological expansion allows a data scientist to query one of our lexicons, without stemming beforehand, and without having to match the stemming standard that we used in our lexicons.

The number of regular expressions for each lexicon can be found in Table 1

Lexicons	ECA	MSA
Abusive Language	310	91
Honorifics	18	36
Pious Expressions	378	96
Polite Expressions		52
Historical Events		91
Targeted Abusive Language		53
Transitional Words and Phrases		658

Table 1: Regular Expression Count of ADST Lexicons

Polite Expression Lexicons: The MSA polite expressions lexicon includes such standard respectful greetings, as أهلاً وسهلاً *welcome*. These expressions were drawn from various MSA dictionaries, including Hans Wehr's Dictionary of modern written Arabic (Cowan, 1994) and al-Mawrid: A modern Arabic-English dictionary (Baalbaki, 1995). The ECA Polite Expressions Lexicon covers polite expressions in ECA and draws from Badawi and Hinds (1986).

Pious Expression Lexicons The MSA Pious Expression Lexicon includes multi-word religious expressions, such as إن شاء الله *God willing* as well as أستغفر الله lit. *I seek God's forgiveness*, which is often used to express disapproval. The primary sources for these expressions are Cowan (1994), Baalbaki (1995), and the Dictionary of Central Asian Islamic terms (Frank and Mamatov, 2002). These expressions are very common in Arabic discourse and can be indicators of polite interactions as well as of religiosity of the speaker. The ECA Pious Expression Lexicon includes similar multi-word religious expressions in ECA drawn from entries in Badawi and Hinds (1986), especially those entries with the label 'Islamic' and 'Christian.'

Honorifics Lexicons The MSA Honorifics Lexicon covers titles used to communicate respect or esteem, such as Sir or Rev. in English. These entries were drawn from Cowan (1994). The ECA Honorifics Lexicon covers Egyptian dialect versions of titles drawn from Badawi and Hinds (1986).

Abusive Language Lexicons The MSA Abusive Language Lexicon includes crude words and multi-word expressions in MSA used to insult. The terms range from mildly insulting يخرب بيتك *May God destroy your house* and ابن كلب *son of a bitch* to stronger vulgarity. These terms were drawn from social media data and online glossaries. This ECA Abusive Language Lexicon includes entries from Badawi and Hinds (1986) that included the labels 'coarse,' 'abusive,' and 'derogatory.' It currently encompasses both targeted and general abusive language (unlike the MSA lexicons which separate the two).

Targeted Abusive Language Lexicons Unlike the more general terms included in the Abusive Language Lexicon, the terms in this lexicon are targeted at specific groups or categories and provide information about both the identity of the speaker and the target of the abuse. For example, the term روافض is a derogatory term for Shi'a Muslims and would not be used by a member of that group. This lexicon also includes dysphemisms referring to specific groups which often play on the structure of the Arabic forms, such as الأمم الملحدة *the atheist nations* for الأمم المتحدة *the United Nations.*

Transitional Words and Phrases Transitional words and phrases are methods of transition in writing, which can connect ideas, demonstrate opposition or contrast, emphasize a previous comment, or demonstrate causation or effect. Features based on

these transitional words and phrases could be used by data scientists for sentiment or emotion classification. Our initial lexicon of words and phrases was in English and leveraged The Writing Center at The University of Wisconsin-Madison (2016). This collection is thematically divided into types of transitions, such as addition, contrast, qualification and many others. The list provided a sufficient foundation from which to begin compiling a more complete list in English and to provide translated equivalents in Arabic. Searches of a number of translation and Arabic language web sources (wordreference.com, Google translate, Quizlet.com) for each term in English led to the Arabic counterparts in meaning. These Arabic counterparts were evaluated for suitability by fluent speakers of Arabic. In several cases, an English term had multiple Arabic counterparts with similar meanings, so these were also added to the list of Arabic terms. Finally, the Arabic list was expanded to include common orthographic variations of the word or phrase, such as removing the short vowels from words, and replacing أ with ا. The script searches for the transitional word or phrase in the text and returns the word/phrase and the frequency of its usage in the text.

4.5 Morphological Features

MADAMIRA is a core component to several of the scripts that make up ADST, because the tool is able to identify the root or dictionary form of the word (lemmatize) and morphologically analyze Arabic words using the word's context in the sentence. While MADAMIRA produces a word by word analysis, the majority of our features are histograms of frequency counts, which could be used by a data scientist as a feature matrix. The ADST scripts which utilize MADAMIRA parse its output and return requested features of a given text. Many of these scripts find and isolate grammatical information and conduct frequency counts for the given text, extracting features for gender, number, person, aspect and clitics (including enclitics and proclitics). Other scripts take advantage of MADAMIRA's lemmatizing feature. A bag of words script returns the frequency of the lemmas in the text, while an out of vocabulary script returns those words in the text that MADAMIRA is unable to analyze. The lemmatizing function of MADAMIRA is also used as a feature to flag the behavior of affixation of the conjunction و waw *and* with the following word. Another script, a variation on the bag of words script, strips diacritics from the stemmed words. Lastly, there is a part of speech (PoS) script, which returns a PoS frequency list from MADAMIRA.

4.6 Spelling and Typographical Features

4.6.1 Elongated words

One hallmark of emotionally expressive informal writing in social media is the use of repeated letters to express emphasis. An English example of this is "LOOOOOL" for "LOL" (laughing out loud). As with direct representations of laughter, we hypothesize that the length of the repetition may correlate with degree of emphasis or intensity of underlying emotion. Therefore, we want to return the length of the repetition as well as the underlying word in its canonical spelling. The component for detecting repeated letters utilizes a wordlist such as Attia et al. (2012) and looks for words not on that list. For every out-of-vocabulary word, it removes repeated letters and then checks if the resulting string is on the list. If it is, it reports the original string, its length, the canonical spelling, and the number of extra letters found. Since it is possible for a correctly spelled Arabic word to have two adjacent instances of the same letter (e.g., حوثيين), only strings of three or more adjacent, identical letters are shortened, and instances where the canonical spelling has either one or two copies of the letter are both considered.

4.6.2 Misspelled words

Since informal writing will generally have more misspelled words than edited text, we also have a component for detecting misspellings. We distinguish two main types of misspellings:

1. Common and commonly accepted spelling variants, such as using an alif character alone instead of a hamza (glottal stop) character seated on an alif, or the using dotless letter variants of taa marbuta and final yaa. (Hypercorrections involving these same variants would also fall into this category.) These are already handled (at least in part) by the regular expressions described in section 4.2.1.

2. 'Major' spelling variants, such as omission or insertion of a letter, a substitution not considered 'minor', or metathesis of adjacent letters.

The former type of misspelling is predicted to be relatively common, and hence relatively uninformative (though its absence may indicate a high level of education, conscientiousness, and/or formality, or use of a device with very accurate auto-correct). The latter type of misspelling may possibly indicate carelessness, fast typing, or even some level of emotional stress. It may also indicate use of a difficult-to-use text input interface. As with the component for finding repeated letters, the component for detecting misspelled words uses a wordlist (currently Attia et al. (2012)) and checks the spelling only of OOV (out of vocabulary) terms. Thus it can only detect non-word errors; real-word spelling errors are out of scope. In order to compile a list of possible intended words, it computes the n shortest paths through a weighted finite-state 'noisy-channel' model, with weights trained on a corpus of quickly typed Arabic. The tool component has a number of parameters that can be adjusted by means of a JSON configuration file, including the wordlist used, the weight (or cost) thresholds for 'minor' and 'major' spelling errors, the number of hypothesized intended words returned, and even the error model used. This component uses OpenFST as a dependency.

5 Summary

The Arabic Data Science Toolkit (ADST) is a tool for data scientists and computer programmers, who may not know Arabic, to quickly and easily extract relevant information from Arabic text, for machine learning applications. At this time, the ADST includes social media features, stylistic features, word category features, and spelling features. Each of these require some underlying Arabic resource for implementation. Our first focus was on author attribution and emotion classification problems, but the toolkit should be useful for other document classification applications as well.
We expect to continue development on this toolkit, extending it with additional features, adding new languages, and improving its ability to scale to large amounts of data.

6 Bibliographical References

References

Attia, M., Pecina, P., Samih, Y., Shaalan, K., and van Genabith, J. (2012). Improved spelling error detection and correction for Arabic. In Martin Kay et al., editors, *Proceedings of COLING 2012*, Bumbai, India.

Attia, M., Pecina, P., Toral, A., and van Genabith, J. (2014). A corpus-based finite-state morphological toolkit for contemporary Arabic. *Journal of Logic and Computation*, 24(2):455–472.

Baalbaki, R. (1995). *al-Mawrid: A modern Arabic-English dictionary*. Dar El-Ilm Lilmalayin, Beirut.

Badawi, E.-S. and Hinds, M. (1986). *A dictionary of Egyptian Arabic*. Librairie du Liban, Beirut.

Cowan, M. J. (1994). *Arabic-English Dictionary: The Hans Wehr Dictionary of Modern Written Arabic*. Spoken Language Services, Urbana, IL.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6(3):169–200.

Frank, A. J. and Mamatov, J. (2002). *Dictionary of Central Asian Islamic terms*. Dunwoody Press, Springfield, VA.

Kulick, S., Gabbard, R., and Marcus, M. (2006). Parsing the Arabic treebank: Analysis and improvements. In *Proceedings of the Treebanks and Linguistic Theories Conference*.

Marton, Y., Habash, N., and Rambow, O. (2013). Dependency parsing of modern standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Moffit, K. and Giboney, J. S. (2015). SPLICE beta v0.9.0. http://splice.cmi.arizona.edu/.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL*.

Pasha, A., Al-Badrashiny, M., Diab, M., Kholy, A. E., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. M. (2014). MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

The Writing Center at The University of Wisconsin-Madison. (2016). Uw-madison writer's handbook. http://writing.wisc.edu/handbook/.