Arabic-SOS: Segmentation, Stemming, and Orthography Standardization for Classical and pre-Modern Standard Arabic

Emad Mohamed
E.Mohamed2@wlv.ac.uk
Research Group in Computational Linguistics, University of Wolverhampton
Wolverhampton, United Kingdom

Zeeshan Ali Sayyed
zasayyed@indiana.edu
School of Informatics, Computing and Engineering, Indiana University
Bloomington, Indiana, U.S.A

ABSTRACT

While morphological segmentation has always been a hot topic in Arabic, due to the morphological complexity of the language and the orthography, most effort has focused on Modern Standard Arabic. In this paper, we focus on pre-MSA texts. We use the Gradient Boosting algorithm to train a morphological segmenter with a corpus derived from Al-Manar, a late 19th/early 20th century magazine that focused on the Arabic and Islamic heritage. Since most of the cultural heritage Arabic available suffers from substandard orthography, we have trained a machine learner to standardize the text. Our segmentation accuracy reaches 98.47%, and the orthography standardization an F-macro of 0.98 and an F-micro of 0.99. We also produce stemming as a by-product of segmentation.

KEYWORDS

orthography standardization, morphological segmentation, classical arabic

ACM Reference Format:
Emad Mohamed and Zeeshan Ali Sayyed. 2019. Arabic-SOS: Segmentation, Stemming, and Orthography Standardization for Classical and pre-Modern Standard Arabic. In *Proceedings of Digital Access to Textual Cultural Heritage (DATeCH '19)*. ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Arabic, like many other morphologically rich languages, suffers from the problem of data sparseness as every lexical unit of the language may show in several forms that may not be easy to map together. In order to combat the data sparseness issue, we often resort to segmentation, a process by which we separate affixes from the stem. Segmentation may be the most important element of an Arabic Natural Language Processing pipeline. In our own experimentation with word embeddings[11], we found out that when we pass raw Arabic to the word embedding program, the top similar words are mostly morphological variants of the same word, which may be useful for some applications but definitely not ideal. When we perform segmentation and stemming, we get semantically related words, just like we do with English. Most morphological segmenters for Arabic are trained on the Arabic Treebank, and are thus good with newswire and probably Modern Standard Arabic in general. However, when we use these tools to segment pre-MSA texts, the quality drops significantly. MADAMIRA [10], the best known Arabic NLP system, which has an accuracy of over 98% on MSA, has an accuracy of 94.7% on a Classical Arabic test set [7]. The main difference between Classical Arabic and MSA lies in their different vocabularies. Moreover, CA's morphology and syntax are more complex than that of MSA [1]. Things are even more complicated by the fact that in Arabic in general, most of the material available uses a sub-standard orthography that maps many groups of characters to one each. In this paper, we present Arabic-SOS, a segmenter, stemmer, and orthography standardizer for pre-MSA Arabic. In *orthography standardization*, which is essential to an Arabic NLP pipeline, we report an *F-1 Macro* score of over 98%, better than any previous system[13]. Our segmenter is the most accurate one so far on pre-MSA Arabic.

The rest of this paper goes as follows: section 2 outlines our solution for pre-MSA segmentation. Section 3 presents our approach to substandard orthography, and section 4 shows how the orthography standardizer helps segmentation. In section 5, we derive stemming from segmentation, and in section 6, we conclude and outline our planned future research.

2 SEGMENTATION

A white space-delimited unit in Arabic is usually more complex than its English counterpart. Many function elements are lumped together with the lexical stem to form a single orthographic unit. As an example, consider the word *>wsyqAblAnkm* (أوسيقابلانكم) and its analysis as shown in figure 1. This word translates into English as '*And will the two of them actually meet you?*' It is made up of the question word >, the conjunction *w*, the future particle *s*, the imperfect verb masculine prefix *y*, the imperfect verb *qAbl*, the dual subject suffix *An*, and the second person, masculine, plural object suffix *km*. While not all Arabic words are as complicated as this example, this demonstrates the importance of segmenting a given word into its various components. In fact, in our training set presented below, two-segment words are the most common (45.3%), followed by three-segment words (33%), one-segment words (14.5%), four-segment words (6.84%), five-segment words (0.3%), and six-segment words (0.01%).

DATeCH '19, May 08–10, 2019, Brussels, Belgium

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/3322905.3322927

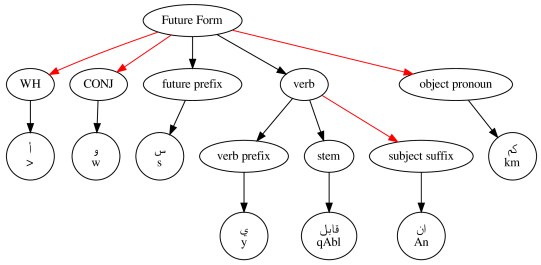

Figure 1: A future form in Arabic. The red lines indicate optional morphemes. WH indicates a question word and CONJ a conjunction. The word is presented in both the Arabic script and the Latinized one-to-one mapping known as Buckwalter encoding

Set	Source	#Words
train_1	Al Manar	85 312
train_2	Al Manar + Classical	141 766
dev	Al Manar	23 786
test_1	Al Manar	24 005
test_2	Classical	5 299

Table 1: Statistics of the datasets used for the experiments

We will first describe the data that has been used in this work in section 2.1, followed by a description of the algorithm and techniques used for segmentation in 2.2. Finally we will discuss the experiments and results in section 2.3 .

2.1 Segmentation Data

We use data from two main sources to build our segmenter: (1) Al-Manar data and (2) Classical Arabic data. Al-Manar[1] is a magazine that was published in Cairo between 1898 and 1935 by Rashid Rida (1865-1935). The magazine aimed at reviving the Arabic language and Islamic sciences and published on various topics from religion to science to politics. The Al-Manar corpus comprises of 132 articles totalling 133 103 words. The Classical Data covers 56 454 words extracted from the Qur'an, Hadith, Islamic law, philosophy and poetry books that date back to the first 4 centuries of Islam. The data was annotated in an iterative fashion using unigrams, bigrams, and trigrams before being passed to a linguist for checking and correction.

We utilize the above two sources of data, to randomly create two training sets, one dev set and one test set. We also include the test set used by [7] as an additional test set to compare the performance of our segmenter with others. This additional test set comprises of 247 sentences and 5 299 words. Table 1 gives a description of the datasets used in our experiments.

[1]http://shamela.ws/index.php/book/6947

2.2 Segmentation Method

There are two methods of performing Arabic word segmentation. In the first method, all possible candidate segmentations of a given word are produced using a morphological analyzer and the most probable segmentation is chosen by ranking the candidate segmentations. The second method treats the problem as a standard sequence to sequence mapping problem. In this type, a sequence of labels is generated for every sentence such that there is a label for every character present in the sentence. Each label in the sequence of labels denotes whether the corresponding character marks the beginning of a segment within a word or is inside the segment or marks the end of the segment.

The first method requires a lexical database and is not robust in handling unknown stems. It does not generalize well to other domains [8]. It has been shown by [12] that sequence to sequence mapping models work well in segmenting dialectical Arabic. In this work, we use a modified version of this second method. In our method, every sentence is mapped to a sequence of binary labels which denote whether the corresponding character is the end of a segment or not, as shown in in figure 2a. This version can produce segmentations equivalent to the ones produced in the original sequence to sequence mapping.

We further reduce the sequence classification problem to a traditional machine learning problem by breaking down the labelling of sequences as the labelling of individual characters independently. We extract contextual features for every character in a sliding window fashion. One step of the sliding window extracts features for that given character according to a chosen feature template and maps it to its corresponding binary label, which indicates whether that character is the end of a segment or not.

2.2.1 Features. The character features are extracted using a template which gives enough contextual information of the character in order to make the classification. A sample of this template can be seen in figure 2b whereas for a complete list of the features used in our model, please refer to table 4. We extract the characters and groups of characters in the neighborhood of the character under consideration along with suffixes and prefixes of the neighboring words. Suffixes and prefixes indicate sequences of letters at the beginning and end of the word and are not used in a strictly linguistic sense.

2.2.2 The Learning Algorithm. We use the method of gradient boosting decision trees (GBDTs) as our core machine learning algorithm in this work. Boosting is an ensemble machine learning technique which works by combining multiple weak models to produce a strong model. These multiple models are learned sequentially such that every model learns from the mistakes of the previously learned models and thus improves upon their mistakes. Gradient boosting machines achieve this be training subsequent models on the residual errors of the previously learned models. This corresponds to minimizing the overall loss function by performing gradient descent in function space.

Gradient boosted decision trees use decision trees as their component model. They have been shown to be very effective and have reached state-of-the-art performance in a number of application areas across machine learning [15], [14], [6]. Xgboost [2] is a fast

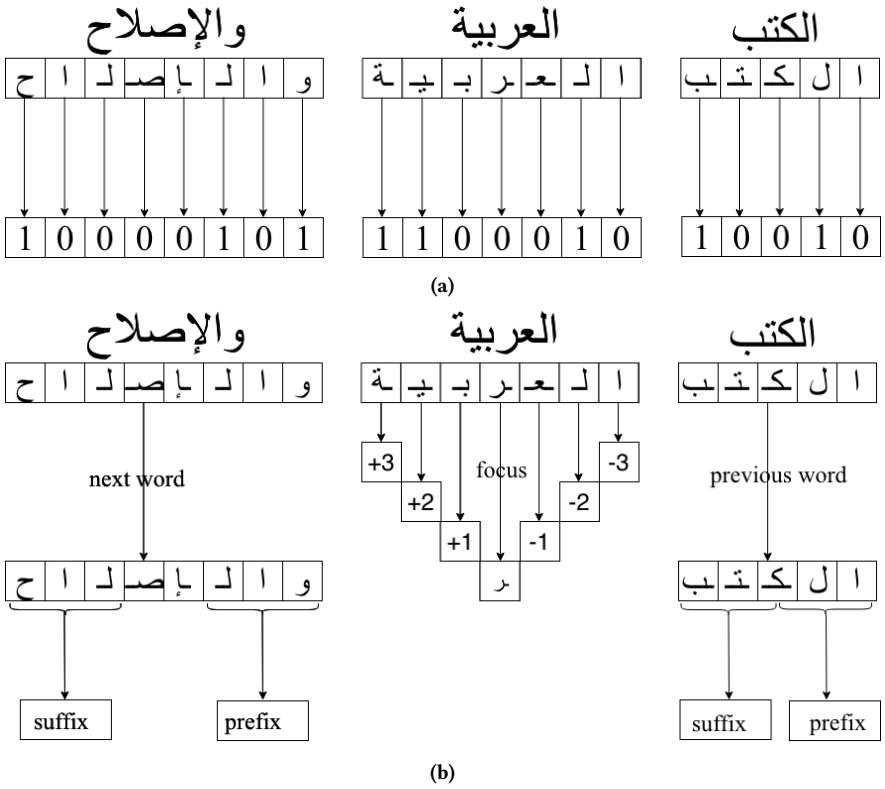

Figure 2: (a) An example of how segmentation of an Arabic sentence can be represented as a sequence to sequence mapping (b) A sample of the feature extraction process for a given character which is marked as focus. The characters in the immediate vicinity of this character are marked as minus1, plus1, etc. Prefixes and suffixes are defined as a group of three characters and are extracted from the surrounding words.

and robust implementation of GBDTs whereas lightGbm [5] and Catboost [3] provide implementations that handle categorical features as input. We experimented with all these implementations and present results with the best performing one in each section.

2.2.3 Evaluation. Individual segments of a word can be easily recovered from the predicted character labels by combining all the characters until an end-of-segment label is found. We adopt word-level accuracy as the evaluation metric for the segmenter. A word is considered to be correctly segmented only if all the segment boundaries have been correctly predicted. There is no partial scoring.

2.3 Experiments

We trained a baseline segmenter model using conditional-random fields [9] with the same feature set as the one used in our core segmenters for the purposes of comparison. While we experimented with many GBDT implementations, we only present the results of our best performing model, which in this case is Catboost. We trained two segmentation models on the two training datasets described in section 2.1. The first model is only trained on training data from the Al-Manar corpus whereas the second model is trained on data from Al-Manar corpus along with that from other classical Arabic sources. Moreover, in order to compare the performance of our segmenter against others, we evaluated Madamira, SAPA[4] and the model described in [7] on our classical test set as well. The results of these experiments are presented in table 2.

	Al Manar	Classical
CRF-Baseline	92.7%	94.96
SOS (Manar)	97.18%	97.17
SOS (Manar + Classical)	97.45	98.47

Table 2: Baselines segmentation accuracy using CRF

System	**Accuracy**
SOS-Manar	97.17
SOS-Manar + Classical	**98.47**
Mohamed (2018)	96.8
MADAMIRA	94.7
SAPA	86.47

Table 3: Comparison with other segmenters on classical test set

2.4 Results and Discussion

Conditional random fields serves as a very good baseline and has been shown to achieve start-art-of-the performance in Arabic segmentation [8]. Our models perform better than CRFs on all test sets. It can be seen that performance of the model improves on both test sets as more classical data is added to the training set. The segmenter performs the best with an accuracy of 98.47% on the classical training set.

Table 3 also shows that our segmenter outperforms other segmenters on the common test set consisting of classical Arabic sentences.

2.4.1 Error Analysis. Basing our analysis on the output of the best scoring experiment (Manar + classical) with the test set being the Classical test set, which has an error rate of 1.53%, we can say that ambiguity, where a single source word has more than one possible segmentation is the main culprit. Examples include *tSdq*, which can be either *tSdq* or *t+Sdq* and *wtbyn*, which could be segmented as either *w+t+byn* or *w+tbyn*.

2.4.2 Feature Ranking. It is useful to know the importance of features in a model as it helps us build better models and has explanatory power that is of great value in linguistic analysis. As is shown in table 4, the most important features are the focus character itself and the characters immediately around it. This indicates that most decisions are made locally, and that the wider context is not as important. This may be understood in one of two ways: (1) most segmentation decisions in our dataset are not ambiguous, and (2) Since we do not have enough lexical context, the importance of the lexical context may not be evident in the model since it does not learn long-range dependencies. We believe the second reason may be more plausible because in our error analysis, most mistakes resulted from words with more than one possible segmentation

Feature	Value	Feature	Value
focus	15.6501	prev_word_suffix	4.2443
next2letters	11.857	chr_position	3.2133
prev2letters	8.8664	minus2	3.1651
focus_word_prefix	7.8821	minus3	2.7478
plus1	7.3599	plus4	2.5857
focus_word_suffix	6.9752	plus5	2.566
plus3	6.7646	following_word_prefix	2.5203
plus2	5.5329	minus4	2.1905
minus1	4.7142	minus5	1.1644

Table 4: Feature importances ranked by the model

The results of segmentation are state-of-the-art, but we need to remember that all the data in this experiment has standard Arabic orthography, which is carefully planned and sometimes unrealistic. Most Arabic data available, be it modern or historic, is not present in the standard form but rather in substandard orthography. In the next section, handle the problem of sub-standard orthography.

3 SUBSTANDARD ORTHOGRAPHY

Arabic orthography has evolved through many stages. This coupled with the fact that Arabic has many dialects has had an impact on the mapping of some sounds to letters. This phenomenon leads to differences in orthography and is called substandard orthography. We will first look at a few common problems in section 3.1 and then describe the way in which we attempt to solve this problem in section 3.2. Finally, we will describe the results of our experiments in section 3.3.

3.1 Types of substandard orthography

There are three major types that we will look at namely, the *hamza* confusion set, the *y* confusion set and the *t/h* confusion set.

3.1.1 The hamza confusion set. One of the most troublesome, and difficult to deal with, sounds of Arabic is that of the *hamza*, which is mostly pronounced as a glottal stop. In the standard orthography, *hamza* can take several forms:

- The form أ has many functions including: (1) being part of the stem, (2) a question word, and (3) an imperfective verb prefix for the first person singular.
- The form إ is usually part of the stem.
- The form آ is a double *hamza* used in assimilated. forms.
- The form ا is a long vowel.

These usually make orthographic minimal pairs. In figure 3, there are three words that mean, from right to left, order, sin, and commander respectively.

أمر إمر آمر

Figure 3: Three forms of hamza forming minimal pairs

Due to the difficulty of the *hamza*, all of these forms are interchangeable in careless Arabic writing.

3.1.2 The y confusion set. The *y* set is confusable word-finally. It has two members as shown in figure 4. The member on the right is pronounced like the English *y*, and is used, among other things, to (1) derive adjectives from nouns, in which case it is treated as a derivational morpheme and is not segmented (e.g. *Haswb = computer, Haswby = computational*), (2) as a first person possessive pronoun, in which case it is segmented (e.g *Haswb = computer, Haswby = my computer*), (3) as a first person pronoun in some special nominative, accusative and genitive constructions and it is also segmented in this case. We can see that this letter is a major form of ambiguity as wrong segmentation has consequences for part of speech tagging and parsing.

ي ى

Figure 4: The *y* confusion set

The member on the left in figure 4 is pronounced as the English *a*, sometimes long and sometimes short, and is usually part of the

stem, and is thus not segmented. In most cases of sub-optimal orthography that we have seen the undotted letter is used instead of the dotted one, although the reverse could also be true. The orthography standardizer determines which one of these is the right one given the context, which helps the segmenter, and any other NLP process, reduce its search space.

Although these two letters have different functions and are pronounced differently, their shape similarity, and the fact that both are vowels, leads to confusion, Furthermore, most of the heritage produced in Egypt in the early history of the printing press used the undotted form for both.

Figure 5: The *t/h* confusion set

3.1.3 The t/h confusion set. The *t/h* confusion set has two members as shown in figure 5: a dotted member and undotted one. In spite of the shape similarity the two forms are completely unrelated. The dotted variant is almost always a singular feminine marker: طبيب = physician while طبيبة = female physician.

The undotted variant, when not part of the stem, is a suffix indicating either a 3rd person masculine singular pronoun or the possessive form thereof. The word طبيبه thus means *his physician*. The form is almost identical to the the one meaning *a female physician*. While the pronominal form must be treated as a separate token for POS and syntactic purposes, this is not usually the case for the feminine marker. The orthography standardizer, given either of these characters occurring word-finally, has to determine whether it is the dotted or the undotted character.

3.2 Handling Substandard Orthography

We approach the problem of standardizing orthography as a multiclass classification problem. A classifier is trained to map every substandard character to its standard form. Given any of the characters in the confusion sets, the task of the classifier is to determine what member of the confusion group it maps to. In order to train the classifier, we extract contextual features for the given character, using the same template we used for the segmenter.

3.2.1 The Data for the Orthography Standardizer. While it is difficult to map from substandard orthography to the standard one due to inherent ambiguity in the process, one can easily map in the reverse direction. Islamweb[2] is a portal containing numerous articles covering many aspects of classical Arabic and Islam relying heavily on the Arabic heritage. Moreover, the editors of Islamweb take utmost care in producing standard Arabic orthography. Hence, we scraped the following data from their portal:

(1) Fatwas: 149299 fatwas totalling 34 650 628 words.
(2) Articles: 1060 articles totalling 1 134 510 words.

We substandardized this data for the purposes of our experiments. We extracted all the substandard characters shown in section 3.1 from this data to create a dataset for the multiclass classification problem. We further divided this dataset into train, dev and test in the ratio of 70:15:15.

[2]http://www.islamweb.net

Class	Class Proportion (in %)	Precision	Recall	F-Score
آ	0.84	0.98	0.96	0.97
أ	9.75	0.98	0.98	0.98
إ	4.09	0.97	0.96	0.96
ا	50.3	1	1	1
ة	6.72	0.98	0.98	0.98
ه	17.43	0.99	0.99	0.99
ى	5.01	0.98	0.98	0.98
ي	5.86	0.98	0.98	0.98
	Macro Average	0.98	0.98	0.98
	Micro Average	0.99	0.99	0.99

Table 5: Class distribution and results in Standardizer experiment

3.3 Experiments and Results

The distribution of classes in the train set is as shown in table 5. This is a heavily imbalanced dataset in accordance with Zipf's law [16]. To ameliorate this problem, we apply random undersampling on the majority class to balance the dataset. Once again, we employed GBDT's to learn the classes. Our best performing model is lightgbm trained with 1500 iterations. It achieves overall accuracy and F-macro of 0.986 and 0.98 respectively.

4 THE EFFECT OF SUBSTANDARD ORTHOGRAPHY

Although standardization is a field of research in its own right, we undertook the task initially as a way to improve segmentation in the real world, and we have created an orthography standardizer that can be used in any Natural Language Processing pipeline. While it is possible to train a segmenter using substandard data, the main task of a standardizer is one of disambiguation, which makes it easier to use other tools in the Arabic NLP pipeline such as part-of-speech taggers, named entity recognizers and parsers.

In order to understand the effect of standardization, we created substandard versions of the train/dev/test sets described in section 2.1. We then trained our segmenter not only on the standard forms, but also on the substandard forms. We also used our standardizer to standardize the substandard test sets thus creating three version of the same test test viz. standard, substandard and standardized. Finally we test both our segmenters on all these three sets. The results of these experiments are summarized in table 6.

From the results it can clearly be seen that the segmenter trained on the standard version of the data and tested on the standard version of test sets achieves the best performance. This can be explained by the fact that there is more ambiguity in the substandard version of text compared to its substandard versions.

Train Set	Train Data Type	Test Data Type			Test Set
		Standard	*Substandard*	*Standardized*	
Manar	*Standard*	**97.18**	94.02	96.88	Manar
	Substandard	88.60	96.70	88.57	
	Standard	**97.17**	93.41	96.79	Classical
	Substandard	89.02	96.83	89.11	
Manar + Classical	*Standard*	**97.45**	94.36	97.12	Manar
	Substandard	88.83	97.22	88.76	
	Standard	**98.47**	96.60	98.17	Classical
	Substandard	90.58	98.09	90.64	

Table 6: Experiment results for studying the affect of standardization

While the substandard version sometimes achieves good results, this may be detrimental in later stages of an NLP pipeline due to the increase in ambiguity. In fact, the standaridized version outperforms the substandard one in most cases, which means it is a recommended step in its own right, let alone its role in disambiguation, disambiguation is inherently harder than in its standard counterpart.

5 STEMMING AS A BY-PRODUCT OF SEGMENTATION

Stemming is the process of reducing a word to its base form, or stem. In a morphologically rich language like Arabic this could be translated into removing all the prefixes and suffixes and maintaining only the lexical unit. In a word like **fs>ETykh**, the segmented form will be *f+s+>+ETy+k+h* where the stem is *ETy*, with the remaining units being affixes. The stemmer takes as input a segmented word and removes these functional units (affixes). This is almost always a straightforward process. Some affixes, however, are ambiguous between lexical and functional units. For example, the unit *hm* could be a word meaning *worry*, or a pronominal suffix meaning *they/them/their*. We use a rule-based system to derive stemming from segmentation. Stemming is at least as accurate as segmentation.

6 CONCLUSION AND FUTURE RESEARCH

We have presented a state-of-the-art pipeline for the pre-processing of pre-Modern Arabic. The data and software will be made publicly available. In the future, we will (1) streamline the whole pipeline in a neural network model, (2) use synthetic data to enlarge our segmentation corpus, which normally requires manual annotation, and (3) apply the methods and the learning algorithm to pre-MSA Named Entity Recognition.

ACKNOWLEDGMENTS

This project was made possible by NPRP grant NPRP10-0115-170163 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] Musaed S Bin-Muqbil. 2006. *Phonetic and phonological aspects of Arabic emphatics and gutturals.* Ph.D. Dissertation.

[2] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[3] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

[4] Souhir Gahbiche-Braham and Helene Bonneau-Maynard. 2012. Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3146–3154. http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[6] Ping Li. 2012. Robust logitboost and adaptive base class (abc) logitboost. *arXiv preprint arXiv:1203.3491* (2012).

[7] Emad Mohamed. 2018. Morphological Segmentation and Part-of-Speech Tagging for the Arabic Heritage. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 17, 3, Article 22 (April 2018), 13 pages. https://doi.org/10.1145/3178459

[8] Will Monroe, Spence Green, and Christopher D Manning. 2014. Word segmentation of informal Arabic with domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2. 206–211.

[9] Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/

[10] Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland.*

[11] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[12] Younes Samih, Mohammed Attia, Mohamed Eldesouki, Ahmed Abdelali, Hamdy Mubarak, Laura Kallmeyer, and Kareem Darwish. 2017. A neural architecture for dialectal Arabic segmentation. In *Proceedings of the Third Arabic Natural Language Processing Workshop*. 46–54.

[13] Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018. Utilizing Character and Word Embeddings for Text Normalization with Sequence-to-Sequence Models. *CoRR* abs/1809.01534 (2018). arXiv:1809.01534 http://arxiv.org/abs/1809.01534

[14] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13, 3 (2010), 254–270.

[15] Yanru Zhang and Ali Haghani. 2015. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies* 58 (2015), 308–324.

[16] George Kingsley Zipf. 1949. Human Behaviour and the Principle of Least Effort: an Introduction to Human Ecology.