Otrouha: Automatic Classification of Arabic ETDs

Eman Abdelrahman
Dept. of Computer Science
Virginia Tech
Blacksburg, VA 24061
emanh@vt.edu

Fatimah Alotaibi
Dept. of Computer Science
Virginia Tech
Blacksburg, VA 24061
falotaibi@vt.edu

***Abstract*—ETDs are becoming a new genre of documents that is highly precious and worthy of preservation. This has resulted in a sustainable need to build an effective tool to facilitate retrieving ETD collections. While there is increasing interest in Arabic ETDs, many challenges exist due to lack of resources and the complexity of information retrieval in the Arabic language. Therefore, this project focuses on making Arabic ETDs more accessible by facilitating browsing and searching. The aim is to build an automated classifier that categorizes an Arabic ETD based on its abstract. Our raw dataset was obtained by crawling the AskZad Digital Library. Then we applied some pre-processing techniques to the dataset to make it clean and suitable for our classification process. We conducted two sets of experiments, i.e., for binary and multiclass classification. In the multiclass classification, we used Support Vector Machines, SVC, Random Forest, and Decision Trees classifiers. Also, an ensemble of the classifiers, that generated the highest accuracy, was built. For the binary classification, we used Random Forest. Additionally, commonly used evaluation techniques such as precision, recall, F1 score, and accuracy were used. The results show better performance for the binary classification with average accuracy 68% per category, where multiclass classification performed poorly with average accuracy 24%.**

***Index Terms*—Arabic ETDs, Arabic Text Classification, Machine Learning, NLP**

I. Introduction

Text classification is the process of categorizing and labeling text based on its content into predefined classes of categories. In the digital age, the amount of textual data is growing rapidly. The exploding number of documents in the digital form makes text classification an important task to help organize documents. Additionally, text classification is known in other real life applications such as spam filtering, news monitoring, and email filtering. There is a variety of text classification algorithms such as Naive Bayes, Support Vector Machine, and K-Nearest Neighbors [1] [2].

ETDs are becoming a new genre of documents that is highly precious and worthy of preservation. Building collections of ETDs is very advantageous for many reasons. First, it helps academic institutes to build their digital libraries and allows them to convey all different formats of content like sound and video. Second, the ease of accessibility makes ETDs available to more users. Third, ETDs help to save space and effort.

Today, there is a rapid growth in the number of Arabic academic institutes and also Arabic researchers, which eventually leads to an increase in the number of Arabic research documents in the scholarly community such as theses and dissertations. For that reason, it is necessary to adapt Arabic ETDs, and encourage Arabic institutes to build their ETD collections in order to make that valuable Arabic scholarly content available to the world. In addition, the available Arabic content on the Web is quite low compared to the number of Arabic speakers. According to W3Techs [3], the estimated percentage of Arabic content is 0.7% of the top 10 million websites on the World Wide Web, as of September 2019, while InternetWorldStats estimates the number of Arabic users to be 5.2% of the Internet users.

To bridge this gap, we present our project "Otrouhs" which is an attempt to help enrich the Arabic content in the Web. The aim is to build an automated classifier that categorizes an Arabic ETD based on its abstract. Our raw dataset are obtained by crawling the AskZad digital library website [4]. Then, we conducted some pre-processing techniques on the dataset to make it suitable for our classifiers. We developed automatic classification methods using Support Vector Machines, SVC, Random Forest, and Decision Trees. We believe this attempt can help make Arabic ETDs more accessible by facilitating browsing and searching. As a consequence, this will encourage researchers, institutes, and universities to make Arabic ETDs available over the Web for more users, and enrich the digital libraries with this body of valuable research.

A. Characteristics of the Arabic Language

The Arabic language is challenging for researchers of natural language processing (NLP). In this section, we present the main characteristics of the Arabic language that makes NLP a very challenging task.

Arabic script is written from right to left with 28 letters. It is significantly different from other languages in the shapes, styles, marks, diacritics, numerals, and distinctive and none distinctive letters. The phonology and spelling of Arabic letters has 28 constants, 6 vowels, and 2 diphthongs [5]. The genders in Arabic language are feminine and masculine. Also, it has singular, dual, and plural numbers. Additionally, there are three grammatical cases in the Arabic language which are normative, accusative, and genitive. The Arabic noun also has the same three cases. The first case is nominative, when the noun is the subject. The second case is the accusative, when the noun is the object of a verb, and last is the genitive case, when the noun is the object of a preposition [5]. Here is a very brief overview of Arabic morphology:

- "Consists of a bare root verb form that is trilateral, quadrilateral, or pent literal" [5].
- "Derivational Morphology (lexeme = Root + Pattern)" [5].
- "Inflectional morphology (word = Lexeme + Features)" [5].

  The features are:
- "Noun specific: (conjunction, preposition, article, possession, plural, noun)" [5].
- "Verb specific: (conjunction, tense, verb, subject, object)" [5].
- "Others: Single letter conjunctions and single letter prepositions" [5].

II. Related Work

Text classification is a challenging research area due to the vast amount of textual data over the Web that needs organizing and monitoring. However, most of the previous work has been conducted on English datasets, while the research on Arabic datasets is extremely limited. In addition, no one, to the best of our knowledge, has attempted to use Arabic ETDs for classification purposes. Different classification models have been used in earlier research for Arabic text classification such as Decision Trees, Naïve Bayes, and Support Vector Machines.

In [12], a performance comparison has been made between Naïve Bayes, K-Nearest Neighbor, and Distance–based, on an Arabic dataset that contains 1000 documents. To compare the accuracy between these classifiers, the author used error rate, recall, precision, and fallout. Different techniques have been applied to preprocessing data such as stop word removal, root extraction, and stemming for dimensionality reduction. The experiment shows that Naïve Bayes outperforms K-Nearest Neighbor and Distance–Based methods.

In addition, [6] presents an automated tool for Arabic text classification (ATC). One goal of that paper is building a representative training dataset that covers different types of text categories, which can be used for further research. Therefore, they used seven different datasets that contain 17,658 texts with more than 11,500,000 words as their corpora. The second goal is making a performance comparison between the SVM and C5.0 algorithms on the same seven different Arabic corpora. In general, the C5.0 algorithm outperformed the SVM algorithm by about 10%.

Gharib et al. [7] used Support Vector Machines, Naive Bayes, K-Nearest Neighbors, and Rocchio classifiers to categorize Arabic text. Then, they compared the results of these classifiers. In order to test the classifiers, they conducted two experiments. In the first one, they used the training set as the test set, while the second experiment involved the leave-one testing method. The Rocchio classifier gives better results when the size of the feature set is small, while SVM outperforms the other classifiers when the size of the feature set is large.

The behavior of N-Gram Frequency Statistics for Arabic text classification was studied by Khreisat [8]. She employed the "Manhattan distance" of dissimilarity measure, and Dice's measure of similarity along with N-Gram Frequency Statistics technique. The experiment showed better classification results for N-Gram Frequency using the Dice measure than the Manhattan measure.

El Kourdi et al. [9] developed an Arabic document categorization tool using Naïve Bayes Algorithm. Non-vocalized Arabic web documents have been classified on five predefined categories. They used 300 web documents as a data set for each category. Also, a cross validation experiment using 2,000 terms/roots have been done. That experiment showed an average accuracy over all categories of 68.78%, where the best categorization performance by category showed 92.8%.

In contrast to the previous literature, [10] shows an experiment on a large Arabic NEWSWIRE corpus without preprocessing. The authors posited that statistical methods are very powerful techniques for Arabic text classification and clustering (maximum entropy). The results show 89.5, 31.5, and 46.61 for recall, precision, and F-measure, respectively, which is a very satisfying result without morphological analysis.

As stated in [11], developing a classifier for Arabic text is difficult due to the complexity of Arabic morphological analysis. The Arabic language has high inflectional and derivational morphology which makes NLP tasks nontrivial. In that research, they used the maximum entropy framework to build a system (ArabCat) that works as a classifier for Arabic documents. According to their results, the ArabCat System shows 80.48 ,80.34, and 80.41 for recall, precision, and F-measure, respectively, while other existing systems such as Sakhr's Categorizer show 73.78, 47.35, and 57.68.

III. The Approach

Our approach steps proceed very much in the same way as previous text classification work. The steps includes data gathering, data preprocessing, building and training the model, and finally evaluating the model.

A. Data Gathering

The dataset to be used in this research was from the United Arab Emirates University Library (UAEU). However, after crawling their website, we could not work on all of the ETDs easily, as many of them were scanned and not native PDF documents. Therefore, we decided to switch to the Saudi Digital Library (SDL), which contains several Libraries such as ProQuest, AskZad, Saudi Cultural Mission in Australia, Dar Almandumah, etc. We chose AskZad since it is rich in Arabic ETDs. According to [4], it is considered as the premier place for Arab academic research.

To collect the data from the AskZad website, we designed a crawler that is able to perform a website scan, log in, and scrape the required data. Fig.1 explains the workflow of our crawling process. Most of the categories in AskZad contain thousands of ETDs. However, we made the crawler download a number of ETD's for each category, after sorting the ETDs descendingly, based on their publication date. However, some of the downloaded ETDs have been excluded since the content was not in Arabic.

Fig. 1. Crawling Workflow.

B. Categories and Data Exploring

We analyzed the AskZad website to gain a deeper understating of their categorization system. AskZad has 16 categories, and the number of ETDs in each category varies. For example, the education category has about 6000 ETDs, where the culture category has about 59 ETDs. All the ETDs are well labeled and categorized according to these 16 categories. Three of the categories were excluded as they don't have enough ETDs. The categories can be found in two places in the AskZad website: in the search options, and as a categorization label for each ETD as shown in Fig. 2.

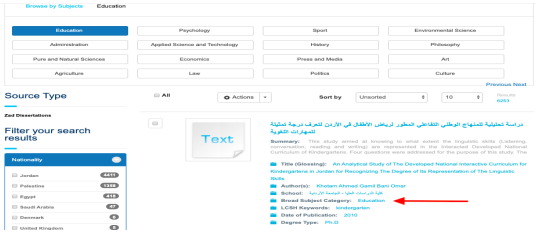

Fig. 2. Sample of Categorized Data.

We excluded three of the categories from AskZad since the number of Arabic ETDs in each of them was very low. The three excluded categories appears in Fig. 3.

We excluded the Sport category too since we did not find a corresponding category in the ProQuest categorization system. The number of categories to work on are 12, consisting of 518 text files with around 124,320 words. The distribution of the documents in our corpus is shown in Table 1.

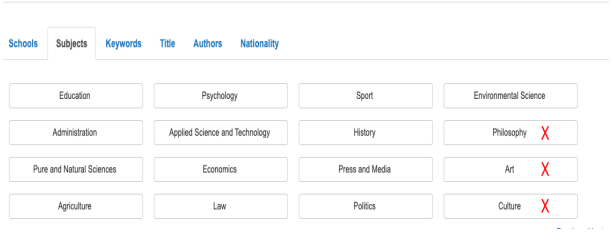

Fig. 3. AskZad Categories and the Excluded Categories.

TABLE I
NUMBER OF DOCUMENTS IN EACH CATEGORY

Category	Number of Documents
Education	50
Administration	50
Pure and Natural Sciences	18
Agriculture	29
Psychology	47
Applied Science and Technology	11
Economies	37
Law	50
History	50
Press and Media	78
Politics	80
Environmental Science	18

C. Mapping

To achieve a greater acceptance, we made the AskZad categories compatible with the ProQuest categorization system. Also, this mapping helps to improve our classification and make it multilabel classification. Since our target is pre-labeled Arabic ETDs, we first obtained AskZad categories in Arabic. Then, we used the English version of the AzkZad website to get an English translation for each category. After that, we did a manual mapping for the translated AskZad categories to the ProQuest subject categories for 2018-2019. For example, we obtained the English translation of the Arabic category, then we mapped it to the corresponding Education category in ProQuest. Fig. 5 shows our mapping of AskZad categories with ProQuest categorization system.

D. Data Preprocessing

Our data set has 12 categories. Each category has titles, abstracts, and keywords for different numbers of Arabic ETDs. We prepared our dataset for the classification experiment by applying some data preprocessing techniques. They involve stop word removal, punctuation marks, and lemmatization.

a) Stop Word Removal: By using a library provided by Natural Language Tool Kit (NLTK), we removed the stop words that are considered not meaningful or are too frequent. For example, we applied stop word removal for an abstract that has originally 204 words, and it turned out then to have 168 words. Table1 shows a list of Arabic stop words.

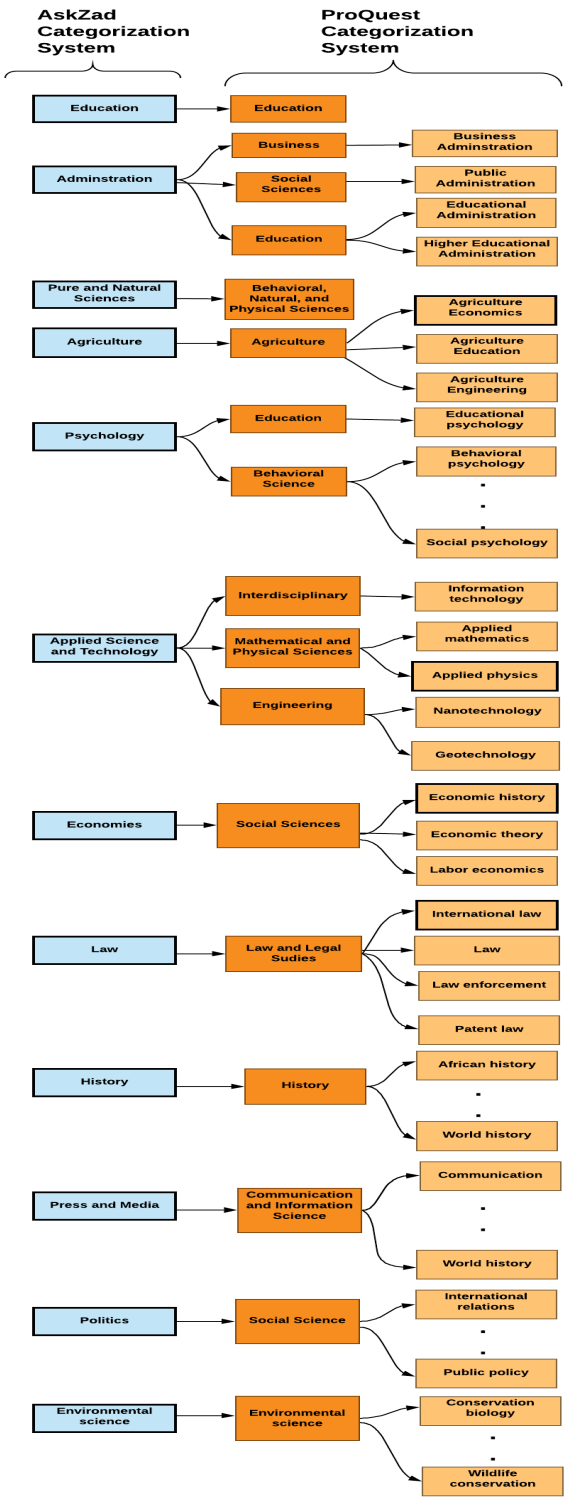

Fig. 4. Mapping AskZad categories to ProQuest categories.

b) Lemmatization Process: The Arabic language has very rich morphology, as roots can be formed to derive a lot of words. For example, a root of a verb can have a prefix, infix, and suffix. Suffixes refer to the time and the tense of the verb. Moreover, the sex of the participant in the verb should be added to the verb whether it is singular or plural [12].

This richness increases the dimensionality of word vectors; therefore, we needed to apply a stemmer that extracts the stem of all the words in each abstract in our dataset. We studied the impact of stemming in Natural Language Processing tasks for Arabic, and throughout our research we found that lemmatization shows more efficiency than stemming particularly in text summarization [13], text indexing [14], and text classification [15]. In the lemmatization process, vocabulary and morphological analyses are used to find the the base form (dictionary form) of a word by considering its inflected forms. On the other hand, the stemmer tries to strip prefixes and suffixes from words to leave a valid stem [16].

Because lemmatization has better impact in text classification, we used the Farasa lemmetizer [17] that outperformed state-of-the-art MADAMIRA and Stanford in Arabic segmentation and lemmatization. We lemmatized each word in all of the abstracts in our dataset by using the Farasa API, then all the lemmatized abstracts have been saved on our local machine for the classification process. In the lemmatization process we used an Advanced REST client tool to make sure the post request is returning valid data.

c) Classification Process: The goal of text classification is to assign documents into a certain number of predefined classes based on their content. There are two types of text classification, which are binary classification and multiclass classification. In binary classification, a document can be assigned in one of only two classes. In contrast, multicalss classification has more than two classes and a document can be assigned to one of these classes. Classification in machine learning is an automatic classification where the algorithms learns from the input data then utilizes this learning to classify new data with more accurate prediction. Text classification in machine learning can be supervised text classification, unsupervised text classification, and semi-supervised text classification. In our project, we used supervised machine learning since each abstract in our dataset is already assigned to a specific category.The aim is to develop automatic classification methods using Support Vector Machines, SVC, Random Forest, Decision trees, and an ensemble classifier that can assign each Arabic abstract in our dataset to the right category. Moreover, we wanted to conduct both binary and multiclass classifications to test the performance of these algorithms for the Arabic ETDs classification task since research in this area is scarce. Fig. 6 shows our classification framework; this framework illustrates the main steps of our classification process.

IV. Experiment and Results

The experiment included training different commonly used classifiers when working with Arabic data to determine how

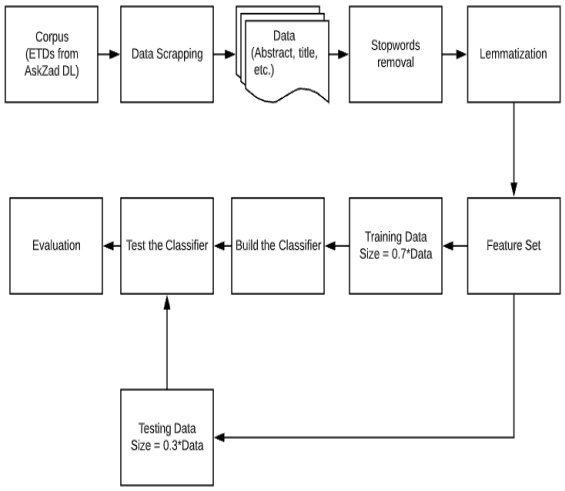

Fig. 5. Classification Framework.

TABLE II
LIST OF ARABIC STOP WORDS

حيثما	وإن	ذوا	عند	أين	هيهات	ذان	والذي	ذي	فيه	ذين	ذات	تلكما
وإذ	نحو	هذين	أن	عل	هاك	آم	فإن	ذو	أما	اللاتي	كذا	متى
ذواتا	نحن	لكم	أم	عسا	يك	ذاك	بهن	منه	بماذا	خلا	بما	لستما
على	كليكما	هنالك	ولا	بعد	لعل	هاتي	إذا	أي	منها	حيث	كم	كان
بها	ذواتي	لذين	هيت	فلا	بيد	أي	هاتان	أو	إليك	كلاهما	وهو	
إذ	اللذين	مع	هاته	وما	إذا	هنا	بهم	كأنما	ومن	هيا		
بذا	ليسا	ذا	هما	يكن	ليت	سوف	به	هذه	ب	كل		
حاشا	قد	فيها	مما	بعض	لذا	يمن	اللتان	تلك	هكذا	فيم		
لولا	كلما	مذ	هذي	بكم	فإذا	مهما	إليكن	أنتما	كلتا	أكثر		
ليس	ثم	يكما	ليسوا	فيما	الذي	كيفما	كليهما	اللواتي	بس	إلى		
ألا	كذلك	كما	لما	أيتما	ذلكم	أي	إي	أنتم	لن	عسى		
لكي	بهما	من	ليست	ذلك	لستا	كأي	ها	وإذا	له	هاهنا		
إلا	شتان	لستم	ذلكن	ذلكما	ماذا	التي	ثمة	أنتن	دون	لهما		
لكيلا	عليك	أنى	كلا	إنه	عدا	لولا	أوه	لهن	لم	ذينك		
لكن	عليه	مه	آها	كي	بل	سوى	منذ	لهم	لك	اللتين		
هم	فمن	كيف	أولاء	هؤلاء	لا	ته	أنا	أيها	والذين	إذن		
إيه	ممن	حتى	هلا	هذا	أولئك	أقل	الذين	غير	لاسيما	تين		
لكنما	ولو	لدى	إنما	لها	بلى	ولكن	لستن	إنما	يا	لكما		
بلى	كيت	جيدا	عن	تلكم	إلى	إليكما	هن	هم	لست	هي		
اللتيا	أنت	ذه	هذان	نعم	هاتين	تينك	ي	هل	بغ	كأين		

each of them will perform. The feature set of each abstract was extracted using CountVectorizer from sklearn. Due to inconsistency between each web page that contains each ETD's information, we faced some issues when extracting the Arabic abstracts. Therefore, we couldn't have an equal representation of each category.

The total number of ETDs collected is approximately 518 including the 12 categories mentioned previously. We split the data into training and testing in the ratio of 70% to 30% respectively. In this study, both multiclass and binary classification were performed. The initial experiment was the multiclass classification, we used Naive Bayes, Decision Tree, Random Forest, and SVM classifiers. As shown in Table III the SVM classifier gave 0.237 accuracy, while the Decision Trees classifier gave an accuracy of 0.244 and the Random Forest classifier gave 0.252. An ensemble classifier consisting of the aforementioned classifiers and Gaussian Naive Bayes gave 0.259. Even though all classifiers performed poorly in our multiclass classification, a slight higher difference is reported in the accuracy of the Random Forest classifier. For that reason, we chose the Random Forest classifier to conduct our second experiment which is the binary classification. We included other evaluation metrics which are precision, recall, and F-measure for our binary classification.

$$Precision: P = \frac{NumberofCorrectClassesFound}{NumberofClassesFound} \quad (1)$$

$$Recall: R = \frac{NumberofCorrectClassesFound}{NumberofCorrectClasses} \quad (2)$$

$$F-measure: F = \frac{2*P*R}{P+R} \quad (3)$$

The results in Table IV shows that binary classification performed much better. Among the twelve categories, eight of them achieved an accuracy of 50% or higher. Two of the eight categories yielded the highest precision, recall, and F-1 measure. These categories are Applied Science and Technology category and the Economics category with the highest accuracy as 95% and 80%, respectively, while the other six categories had satisfactory results for precision, recall, and F-1 measure. The remaining four categories got less than 50% accuracy.

TABLE III
RESULTS OF MUILTICLASS CLASSIFICATION USING DIFFERENT MODELS

Classifier	Accuracy
SVM	0.237
Decision Trees	0.244
Random Forest	0.252
Ensemble Classifier	0.259

V. CHALLENGES

A. Availability of Arabic ETDs

At the beginning we faced a major challenge finding Arabic ETDs on the web. Most of the easily accessible ones are just unstructured Arabic ETDs. There are some Arabic digital libraries that contain good Arabic ETDs in terms of the amount and labeling. However, these digital libraries are geo-restricted content that can not be accessed from the USA.

B. Ability to read the content

Initially, we inspected the UAE university website, and we studied the website structure, so we can build a crawler that can download their content of Arabic ETDs. After that, we successfully designed a crawler and finished scraping the required data. However, when we tried to extract the text from the PDFs, we found that most of the PDFs were scanned and not native PDFs. To solve this issue, some OCR tools have been used such as ABBYY FineReader, Enolsoft, and CISDEM. However, none of them has worked, as they produced unreadable output. Therefore, Fatimah had to request a

TABLE IV
RESULTS OF BINARY CLASSIFICATION USING RANDOM FOREST

Category	Precision	Recall	F-Measure	Accuracy
Education	0.69	0.66	0.65	0.66
Administration	0.18	0.45	0.26	0.36
Pure and Natural Sciences	0.62	0.63	0.62	0.63
Agriculture	0.69	0.65	0.64	0.66
Psychology	0.21	0.5	0.30	0.43
Applied Science and Technology	0.95	0.95	0.95	0.95
Economies	0.8	0.79	0.79	0.8
Law	0.57	0.57	0.53	0.53
History	0.43	0.43	0.39	0.4
Press and Media	0.42	0.42	0.42	0.46
Politics	0.3	0.37	0.37	0.6
Environmental Science	0.79	0.60	0.54	0.63

Mubtath account related to the Saudi Arabia Cultural Mission from the Saudi Digital Library (SDL) that allowed us to access their website. We explored all of the available Arabic digital libraries found in SDL and found that the AskZad library is the only one that provides abstracts as metadata.

C. Encoding

The encoding was one of the challenges that prevented us from being able to store the data in a readable way. This is because not all the NLP tools support languages other than English.

VI. FUTURE WORK

We intend to increase the size of the corpus by scraping more data. We believe this corpus is representative, and will be of particular interest for researchers willing to work in the area of Arabic ETDs for further investigation and bench-marking in NLP. We will also run each classifier against both Arabic and English abstracts separately. This will help us determine how each classifier performs on the same data but in different languages. To improve the accuracy, we will try to use word embeddings as used in [18].

ACKNOWLEDGMENT

We would like to deeply thank Dr. Fox for his continuous support. Also, our colleague Palakh Jude provided guidelines and assistance. Special thanks go to ther Saudi Digital Library for giving an account for Fatimah Alotaibi, which made this project possible. Related funding was provided through IMLS grant LG-37-19-0078-19.

REFERENCES

[1] Rasha Elhassan, Mahmoud Ahmed (2015), "Arabic Text Classification review" International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 1, January 2015.
[2] "Languages Used on the Internet." Wikipedia, Wikimedia Foundation, 7 Oct. 2019, en.wikipedia.org/wiki/Languages-used-on-the-Internet.
[3] ArabianBusiness.com. (2019). Arab internet users forecast to rise to 226m by 2018. [online] 10 Oct. 2019, arabianbusiness.com/arab-internet-users-forecast-rise-226m-by-2018-626635.html.
[4] Askzad.com. (2019). [online] Available at: http://www.askzad.com/ [Accessed 4 Oct. 2019].
[5] Odeh, A.; Abu-Errub, A.; Shambour,Q.; and Turab, N. (2014). "Arabic text categorization algorithm using vector evaluation method." International Journal of Computer Science & Information Technology, 6(6), 83-92.
[6] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, S., and AlRajeh, A., "Automatic Arabic Text Classification" 9th International journal of statistical analysis of textual data, pp. 77-83, 2008.
[7] Gharib, T., Habib, M., and Fayed, Z., "Arabic Text Classification Using Support Vector Machines" International Journal of Computers and Their Applications, Vol (16), Issue(4), 2009.
[8] Khreisat, L., "Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study." In proceedings of the 2006 International Conference on Data Mining, DMIN 2006. Pages 78-82. 2006.
[9] Mohamed El Kourdi, and Amine Bensaid, and Tajje-eddine Rachidi. "Automatic Arabic Document Categorization Based on the Naive Bayes Algorithm." In proceedings of the COLING2004 Workshop on Computational Approaches to Arabic ScriptBased Languages, Switzerland. Pages 51–58. 2004.
[10] Sawaf, H., J Zaplo,J., and Ney, H., "Statistical Classification Methods for Arabic News Articles." In Proceedings of the ACL/EACL 2001 Workshop on Arabic Language Processing: Status and Prospects, Toulouse, France. 2001.
[11] Alaa M. El-Halees. "Arabic Text Classification Using Maximum Entropy." The Islamic University Journal 15(1). Pages 157-167. 200
[12] Duwiri, R., 2007. "Arabic Text categorization." Int. Arab J. Inform. Technol., 4: 125-131.
[13] T. El-Shishtawy and F. El-Ghannam, "A Lemma Based Evaluator for Semitic Language Text Summarization Systems." ArXiv Prepr. ArXiv14035596, 2014.
[14] F. K. Hammouda and A. A. Almarimi, "Heuristic Lemmatization for Arabic Texts Indexation and Classification." 2010.
[15] R. KOULALI and A. MEZIANE, "Experiment with Arabic Topic Detetction." J. Theor. Appl. Inf. Technol., vol. 50, no. 1, 2013.
[16] H.Mubarak. "Build Fast and Accurate Lemmatization for Arabic." 2017.
[17] Abdelali A., Darwish K., Durrani N., and Mubarak H. "Farasa: A Fast and Furious Segmenter for Arabic." Proceedings of NAACL-HLT, San Diego, California (2016).
[18] Soliman,A. Eisa, K., and El-Beltagy,S. "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP", in proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, UAE, 2017.