

Automatic Classification of Arabic ETDs

Eman Abdelrahman and Fatimah Alotaibi

Supervised by: Dr. Edward Fox

CS6604 Course Project

December 10th, 2019

Virginia Tech, Blacksburg, 24061

This document has our additional work about the “**Otrouha: Automatic Classification of Arabic ETDs**” research project.

Data gathering:

- ProQuest and ND LTD.org

Initially, we looked for Arabic ETDs in the popular ETD digital libraries such as ndltd.org and ProQuest. We found that the ones tagged with the Arabic language were not really Arabic; some of them were Farsi or other different languages.

- UAEU

Then we chose United Arab Emirates University “Scholarworks @ UAEU” to collect Arabic ETDs. We inspected their website and found that it is highly crawlable, as shown in the figure below. Accordingly, we designed a crawler that can download the required data. Their digital library contains 75 dissertations and 687 theses. The ETDs that contained Arabic translation of their abstracts, keywords, and title are about 300 ETDs including different disciplines. Once we downloaded the ETDs, we harvested the metadata corresponding to each ETD and saved them as JSON files. The following is an example of the metadata harvested.

Table 1. Metadata of UAEU

The metadata contains the field of “Department” which is the ETD’s class. Since we were planning to be using the ProQuest categorization system, we mapped the Scholarworks@UAEU categories to ProQuest categories. For the categories that don’t exactly match, we chose the nearest discipline. The following figures 2 and 3 show examples of the UAEU and ProQuest Architecture category and its subcategories.

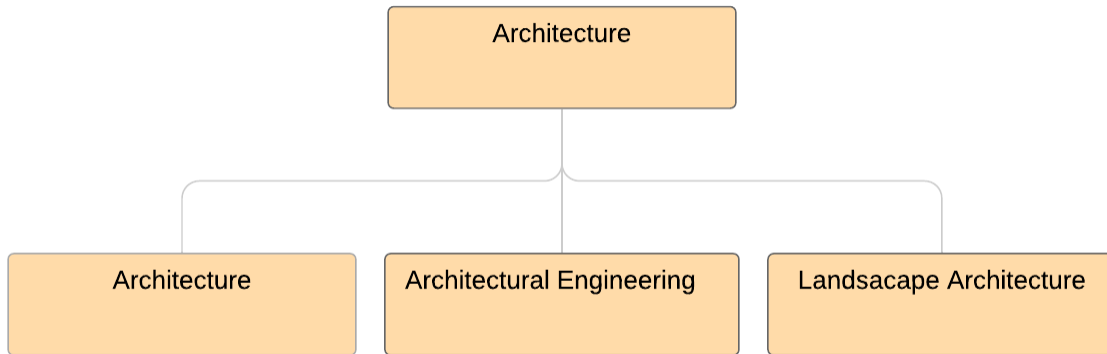


Figure 2. ProQuest Category of Architecture

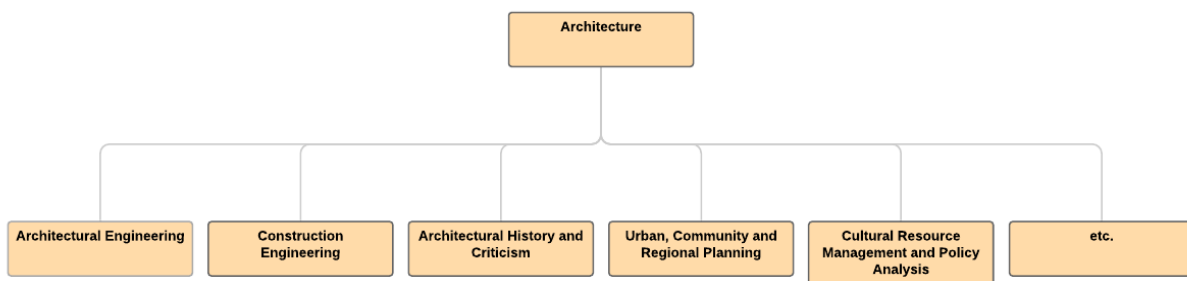


Figure 3. UAEU Category of Architecture

Once we downloaded a number of ETDs to work on, we found that most of them are not native PDFs, but rather scanned PDFs as shown in Figure 4, which made extracting the text not straightforward.

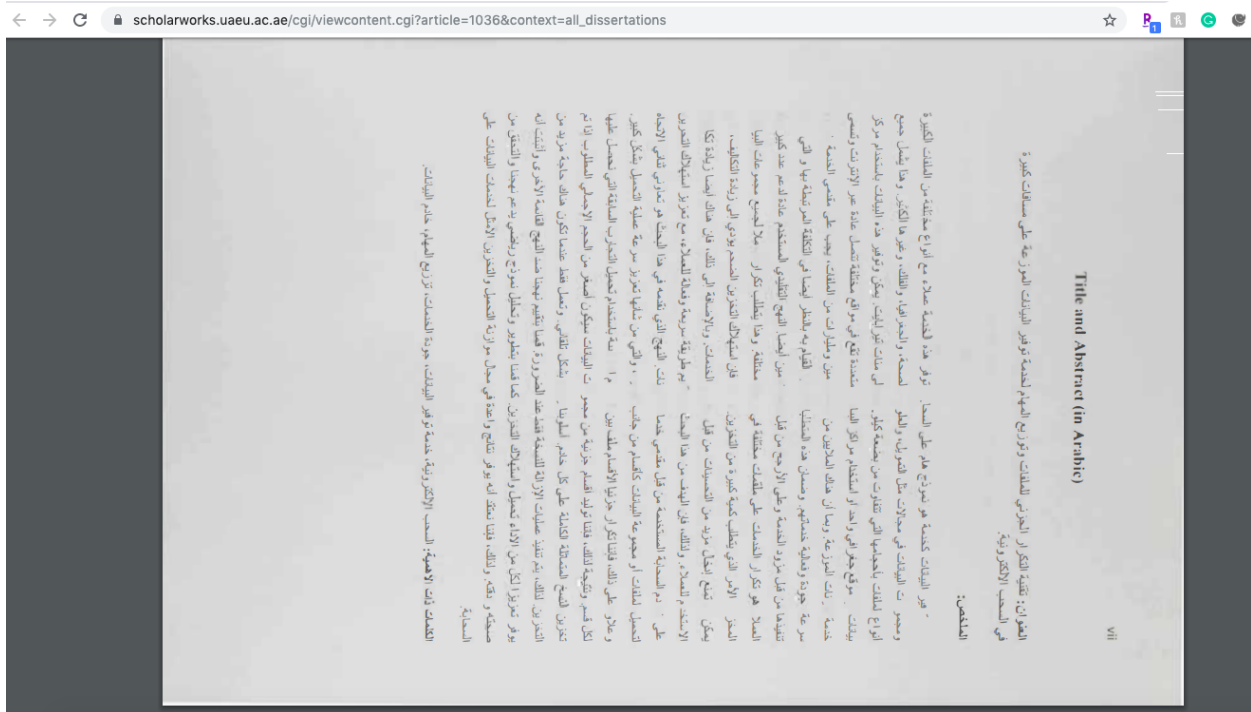


Figure 4. Example of scanned PDF from UAE library

Arabic OCRs tools

To solve the aforementioned issue and extract the text, we tried different Arabic OCR tools such as ABBYY FineReader, Enolsoft, and CISDEM. Unfortunately, none of these tools has worked. They convert characters incorrectly and give unreadable output as shown in Figure 4. So we decided to use another source for Arabic ETDs. There are some other Arabic digital libraries that contain Arabic ETDs in terms of the amount and labeling. However, these digital libraries are geo-restricted so can not be accessed from US.

We found AskZad that is part of the Saudi Digital Library because it has a large number of Arabic ETDs and most importantly, it provides the Abstract as metadata in their webpage.

Create student account at Saudi Digital Library

In order to have access to the Saudi Digital Library, you need to have a student account. Creating an account there has many restrictions. The user has to be a Saudi student who is enrolled in one of the Saudi universities, or a Saudi student who studies abroad. Fatimah asked for an “Mubtath” account through the “Saudi Arabia Cultural Mission”. This process took a long time which held us back from making progress in our project.

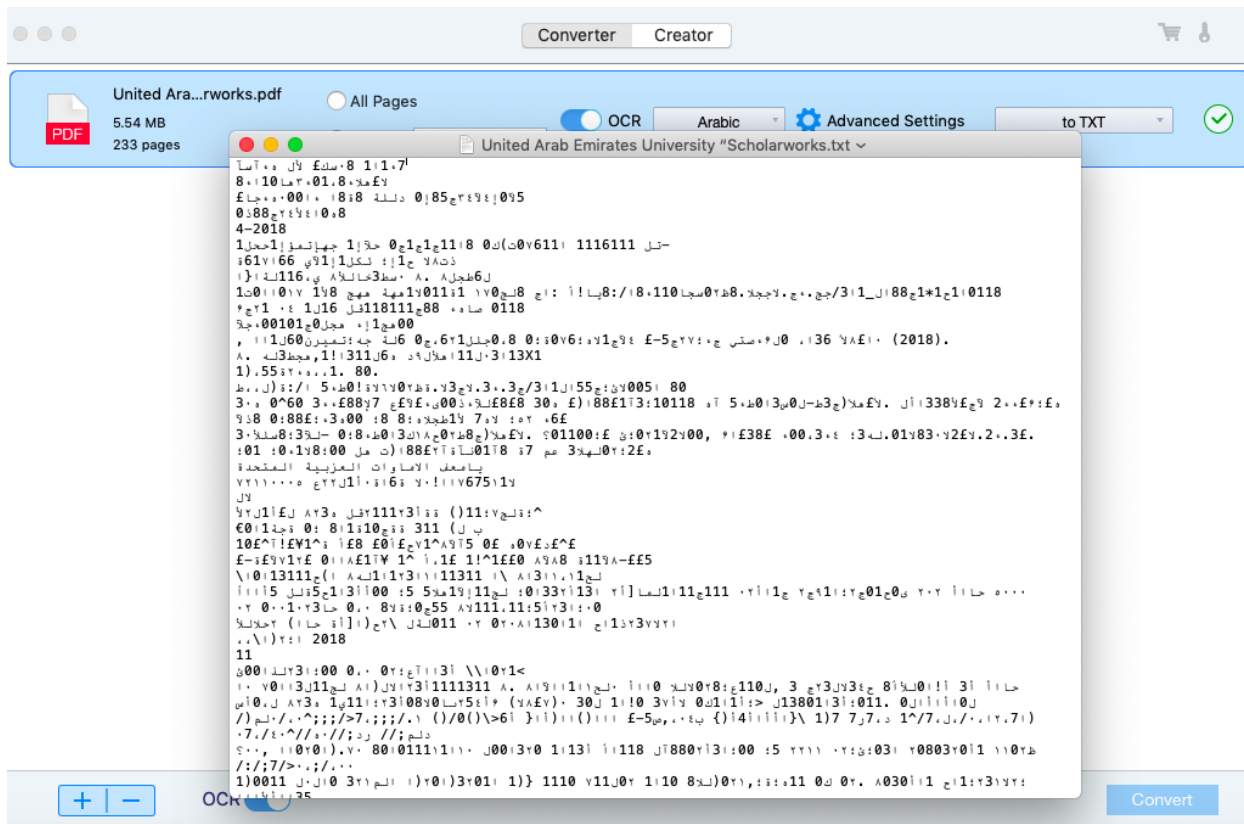


Figure 5. The output of CISDEM OCR tool

- AskZad

There were two challenges when crawling the AskZad Digital Library website. First, the website's crawlability is very low. Therefore, we had to use the full XPath of each element we are looking for. While revising the data, we found that some of the data contains a lot of noise due to the change of the website structure. This leads to getting elements that are not what we are looking for. For example, getting the Advisor Name instead of getting the Abstract. The second challenge was that the Crawl Rate Limit is 45, so among the ETDs downloaded within that limit, some of them weren't what we are looking for. These two challenges significantly reduced the data size.

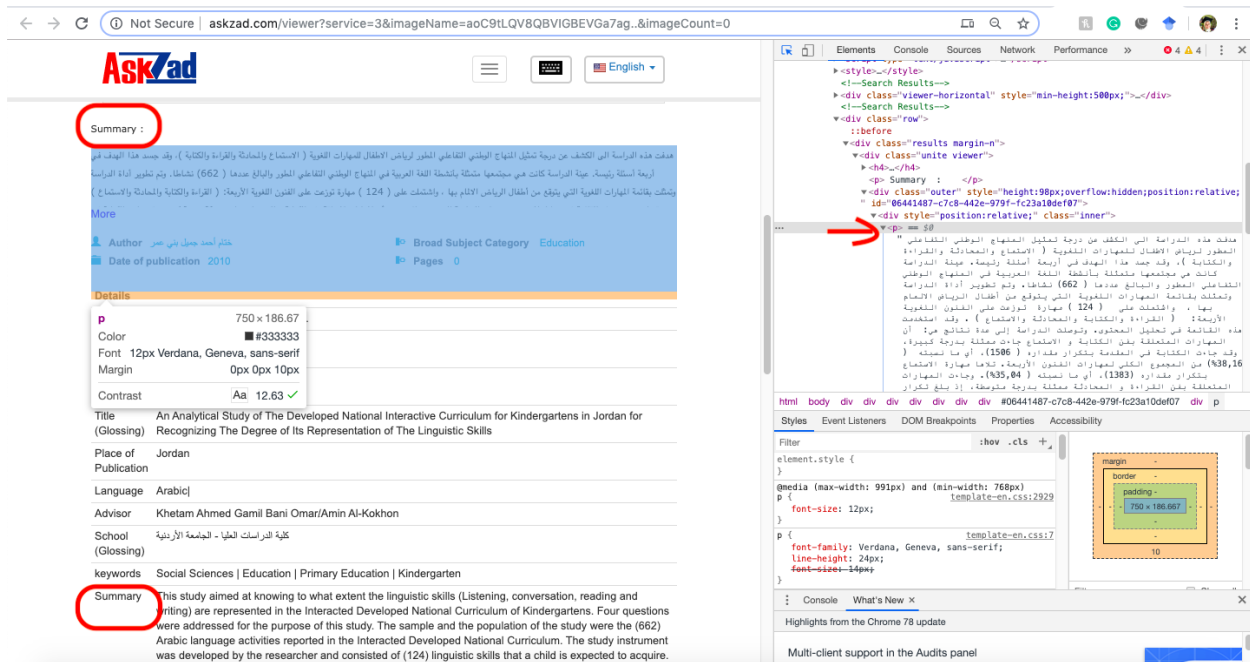


Figure 6. Inspection of Askzad page

Data Preprocessing

Stemming vs. Lemmatization

First, we used different stemmers to preprocess our data. We gave the same input text for these stemmers and each of them provided a different output based on the way each of them stems words. Also, these stemmers give the stem of the word which most of the time changes the meaning of the context. Throughout our research we found that lemmatization shows more efficiency, especially in the Arabic language, since it is a highly inflectional and derivational language. As a result, we chose the FARASA lemmatizer.

Figures 7, 8, and 9 are illustrative examples of the difference between stemming and lemmatization in the Arabic language.

/ Arabic Language Technologies / ALT Server / Demos / Farasa

Please enter your text:

أدخل النص المراد معالجته:

يُشار إلى أن اللغة العربية يتحدثها أكثر من 422 مليون نسمة ويتوزع متحدثوها في المنطقة المعروفة باسم الوطن العربي بالإضافة إلى العديد من المناطق الأخرى المجاورة مثل الأمازغ وتركيا وتشاد والسنغال وإريتريا وغيرها. وهي اللغة الرابعة من لغات منظمة الأمم المتحدة الرسمية الست.

Please note that there are some limitations to try the Dependency Parser:

- The demo is confined to process only three sentences per request, each sentence shouldn't exceed 20 words.
- The length of the text to be processed should be within 400 characters.

Lemmatization أصول الكلمات

Process text معالجة النص

Clear text مسح النص

Text length 269 عدد أحرف النص

أشار إلى أن لغة عربي تحدث أكثر من 422 مليون نسمة توزع متحدثوها في منطقة معروف اسم وطن عربي إضافة إلى عديد من منطقة آخر مجاور مثل أهواز تركيا تشاد سنغال أريتريا غير . هي لغة رابع من لغة منظمة أمة متحد رسمي ست .

Figure 7. Result of lemmatization by Farasa

Type ONE word, select language and press "Stem!" button.

غيرها. وهي اللغة الرابعة من لغات منظمة الأمم المتحدة الرسمية الست arabic Stem!

يشار إلى أن اللغة العربية يتحدثها أكثر من 422 مليون نسمة ويتوزع متحدثوها في المنطقة المعروفة باسم الوطن العربي بالإضافة إلى العديد من المناطق الأخرى المجاورة مثل الأمازغ وتركيا وتشاد والسنغال وإريتريا وغيرها وهي اللغة الرابعة من لغات منظمة الأمم المتحدة الرسمية الست

Figure 8. Results of Snowball Arabic Stemmer

Assem's Arabic Light Stemmer (BETA)

Description

Welcome to the Arabic Light Stemming Algorithm made for [Snowball](#), it's fast and can be generated in many programming languages (through Snowball).

Demo

Type some Arabic text and press "**Stem!**" button or "**File**" to read from a local ".txt" file



يُشار إلى أن اللغة العربية يتحدثها أكثر من 422 مليون نسمة ويتوزع متحد

STEM! **FILE**

Stats words: 43 stems: 38 ratio: 1.13

يشار الي ان اللغ عرب يتحدث أكثر من مليون نسمة يتوزع متحد في منطق معروف 422
باسم وطن اضااف عديد مناطق اخري مجاور مثل اهواز تركي تشاد نغال ارييري غير وه
رابع لغا منظم امم متحد رسم الست

Figure 9. Results of stemming by Assem

During our work with the FARASA API, we needed to send a post request and make sure that it sends valid data back every time. To do that, we used the Advanced REST client tool. Figure 10 shows how this tool works.

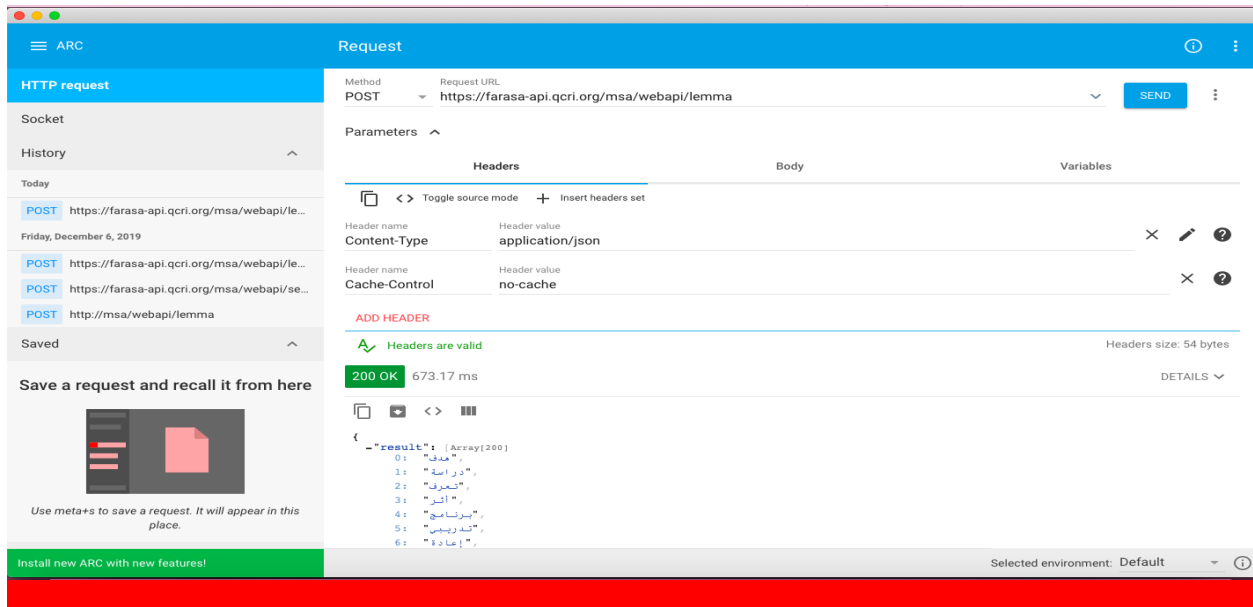


Figure 10. Rest client tool connected with FARSA

Classification Process

After we prepared clean data, we ran the SVM model for training. However, the performance was very low.

We tried the tree decision classifier and there was no significant difference.

In this first experiment which is multi-class classification, none of the used models gave a good accuracy. We finally tried Random Forest, which gave us a slight higher difference, so we decided to choose it for a second experiment which is binary classification.

In the binary classification, Random Forest gave a satisfactory results based on the recall, precision, and F-1 measures, and the accuracy. Finally, we think that working on larger dataset will make the accuracy better and the results more credible. Therefore, we want to increase our corpus for further investigation and benchmarking. Also, we aim to make this corpus available

for researchers who want to pursue the work on Arabic ETDs. To do that, we will use [Sketch Engine](#). This engine helps building corpora in different languages.

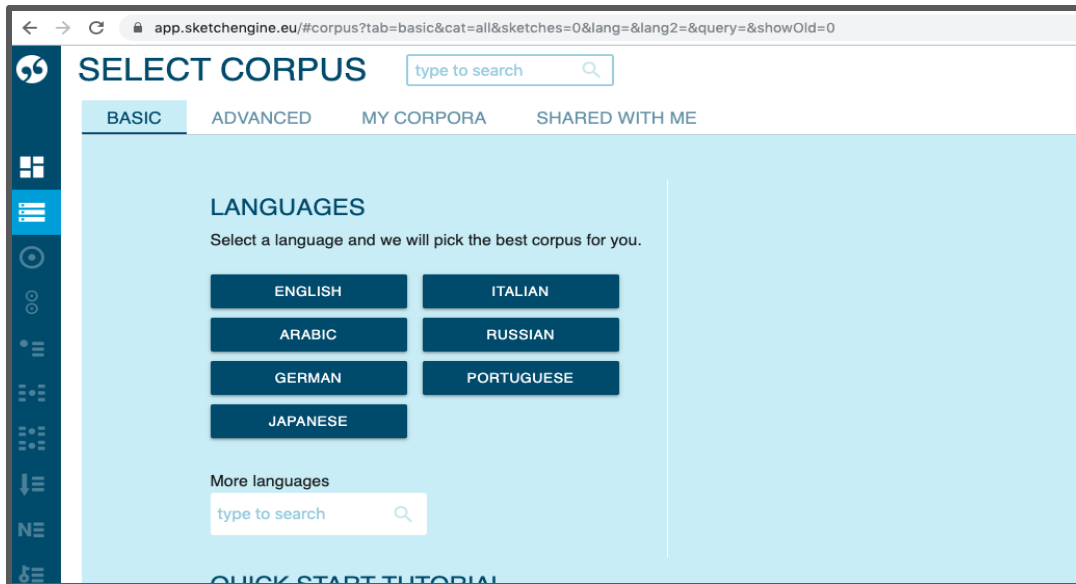


Figure 11. Sketch Engine