Building Datasets for Dialect Classifiers using Twitter

Anonymous ACL submission

Abstract

This paper outlines a scalable approach to creating manually annotated dialectal data sets from Twitter, at little to no cost, intended for a variety of machine learning models that may rely on dialect identification. We distinguish between two Arabic dialects, Egyptian and Gulf, using a variety of dialectal terms, and store an evenly balanced corpus of 300 thousand tweets. Experimental results show that benchmark classifiers trained using the approach outlined in this paper can determine in which dialect a given sentence was written with impressive accuracy, further revealing an important semantic and social connection.

1 Introduction

There are nearly 500 million Arabic speakers in the world, making it the largest member of the Semitic language family. Arabic is also the fourth most used language on the Internet with over 175 million users (Internet World Stats, 2017), and yet despite its significant influence, the Arabic language has garnered comparatively little attention in modern computational linguistics (Stanford NLP Group, 2017). The majority of published Arabic is in Modern Standard Arabic (MSA), which differs substantially from regional dialects (Zaidan, 2014). Aside from the speech domain, linguistic research tends to heavily rely on MSA data sets (Zaidan, 2014). Progress has been made to create Arabic dialectal data sets that can be used to train and evaluate classifiers for dialect identification. Such classifiers have also been found to outperform baselines that are solely reliant on MSA (Zaidan, 2014) and can be used for speech recognition systems, machine translation, personalizing search engine results and digital advertising. Dialectal Arabic is more often present in informal settings such as blogs, comments, forums, and social media. The Arabic On-line Commentary Data Set (AOC) (Zaidan, 2014), and the Cross Lingual Arabic Blog Alerts (COLABA) project (Diab et al. 2010), are efforts to derive dialectal Arabic from such sources. As far as we know, there are no manually annotated Arabic dialectal data sets created entirely from social media sources. “In terms of language on Twitter, tweets in Arabic account for over 72% of all tweets generated in the Arab world.” (Arab Social Media Report, 2017).

We propose a novel way to build dialectal data sets from Twitter that can be used to train highly accurate models to determine in which dialect a given tweet was written. Using the Twitter API, we devise a flexible approach to identify, collect, and annotate tweets primarily written in two Arabic dialects, Egyptian and Gulf. Our method does not incur a huge cost and can be used to annotate thousands of tweets at a time. The approach utilizes non-polysemous, dialectal keywords to avoid capturing tweets containing terms that are spelled similarly but pronounced differently, or mean different things across dialects, thereby improving the likelihood of a collected tweet belonging exclusively to the dialect of interest. We initially conduct exploratory data analysis (EDA) and score tweets based on lexical diversity, analyzing the results to identify users more likely to frequently use relevant dialectal terms. We then inspect the resulting user timelines to collect up to 3,000 tweets (the API’s rate limit) and annotate them in the process. Once we have a large enough sample of annotated tweets, we combine the Egyptian and Gulf data sets into a single corpus and conduct topic modeling using Latent Dirichlet allocation (LDA) to reveal linear combination of terms that characterize or separate the two classes, visualizing the various topics through the tool

LDAvis. With Egyptian and Gulf as the response variables (classes) and the collected tweets (documents) as the predictor variables, we finally use a benchmark Naive Bayes on a test set of 75,735 manually annotated tweets to achieve a dialect prediction score over 92%.

Building the corpus

Using the Twitter Streaming API, we initially filter real-time tweets using a region specific keywords list created for each dialect, Egyptian and Gulf, referred to as EG and GLF.

- GLF terms for Hijazi, Najadi, Omani, Kuwaiti, Emirati, Bahraini.
- EG terms for various regional dialects, such as Alexandrian, Sa'idi, Isma'ili, Badawi, Cairene.

Code from the book *Mastering Social Media Mining with Python* (Bonzanini, 2016), was used to stream tweets from the Twitter API. The dialectal keyword lists were created from a selection of terms available on *Mo3jam.com*, and additionally solicited from Native speakers of those dialects.

Polysemy in the Arabic Dialects (Ibrahim, 2005) was used to research terms that co-occur in various Arabic dialects and have different meanings. This was part of an ongoing effort to separate the EG and GLF sub-corpora as much as possible, by avoiding polysemous terms when streaming tweets to ensure a more truthful representation of the dialect being queried. Furthermore, collected tweets were inspected for homographs during EDA, in order to ensure sufficient distance between the sub corpora before the manual annotation of the two classes.

EG	GLF
السكه (el sekkah)	الطريج (el tareeg)
ازيك (ezayek)	خربز (kharbez)
ضوافر (dawafer)	ازبن (azben)
مناخير (manakheir)	ابخص (abkhas)
كوباية (kobaya)	شخاط (shkhat)
أزاي (ezay)	البشكاره (el bashkarah)
تهييس (tahyeis)	مغسلة (maghsalah)

Figure 1: Sample of non-polysemous terms used to query dialectal tweets for each class.

After collecting at least 100 tweets from each keyword-filtered stream, we identify users that contribute the most tweets to those streams, as well as users with high lexical diversity scores. A simple function finds the length of unique terms in the text and divides it by the length of the entire text, and is used to mask tweets with lexical diversity scores over 0.4, while an additional mask is used to exclude tweets under 40 characters.

	cleaned_name	cleaned_text_score	cleaned_text	cleaned_text_len
7	h_sawires	0.411765	أزاي الناس كانت بتفرد في الحسينات قبل إختراع تويتر	51
44	titoozahran	0.410714	مش عارفة هصبي الساعه عشان احضر مكنش أزاي د الساعه دلوقتي	56
49	AbdelaZizo_	0.431818	أنت غيرت المقلم كدا أزاي الشيخ ماشاء الله _	44
56	Ejal3mril0	0.426230	اللي شكله قايل ما الشوفك عيبًا غفر ضايع بحسوه أزاي عيبًا	61
73	mariam_red	0.511628	أزاي تعرفت علي افضل صديق ؟ عندك ي؟ – جامعه	43

Figure 2: From left to right (Twitter username, lexical diversity score, tweet content, tweet length).

To ensure that the tweets truthfully represent the dialect being annotated, we manually inspect a user's timeline based on simple criteria before adding their tweets to the sub-corpora. A user's own tweets (excluding retweets and mentions) should be heavily inundated with dialectal terms and should not contain too many retweets or spam-like behavior. Users timelines should also not contain high volumes of repetitive tweets or tweets that are cross-cultural.

For example, user timelines with high volumes of religious verses are avoided, as similarity in this content across dialects adds noise to the data. Prioritization was given to user timelines with tweets focused on specific topics, such as fashion, politics, humor, parenting, romance, and sensitive subjects. We then conduct a Timeline Search using Twitter's REST API and collect the most recent 3,200 tweets (rate limit for the API) from qualified users. Additional users whose tweets appear on those chosen timelines can also often be highly dialectical, and upon further inspection can be added to the timeline search.

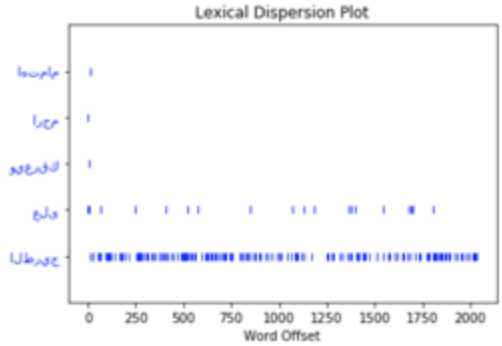

Figure 3: Plotting of the GLF subcorpus using EG and GLF terms reveals that GLF terms have a much higher lexical dispersion, further confirming the lack of impurity in the different subcorpora.

We check for GLF terms in the EG sub-corpus and vice versa by exploring them in a Lexical Dispersion plot. In Figure 3, low word offset for EG terms in the GLF sub-corpus is a way of confirming whether or not the classes are sufficiently separated.

2 Text analysis and visualization

After manually annotating the data according to dialect class, we conduct EDA on the entire corpus.

- Corpus size: 300k (evenly split and labeled EG/GLF).
- Vocabulary size: 356,726

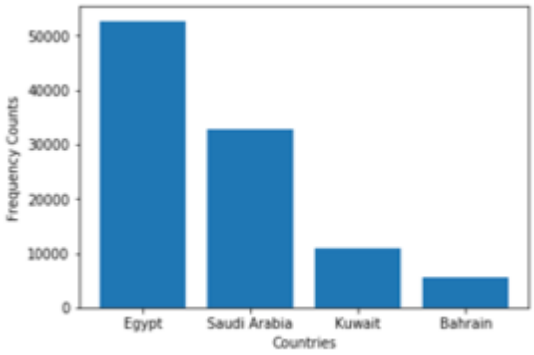

Figure 4: The geographic origins of tweets show a balance between classes (i.e.: The number of tweets originating from Egypt is roughly equal to all the Gulf countries combined).

Dialectal terms found during EDA but not included in the initial keywords list were used to gather more tweets from the real-time streamer. Alternatively, polysemous terms appearing in high frequencies can be added to a stop words list before vectorizing the data.

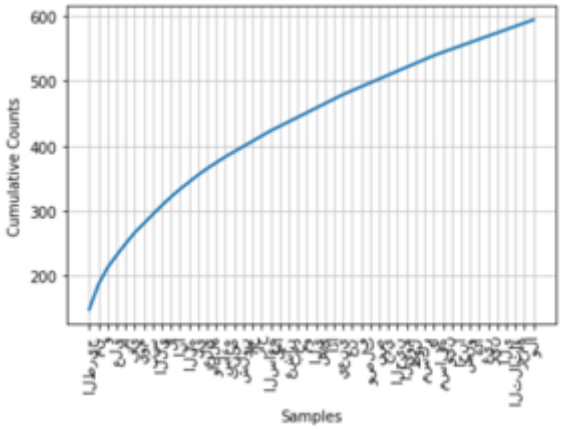

Figure 5: By determining the most frequent terms in our corpus, using the NLTK library we can determine the cumulative value counts of said terms.

3 Topic Modeling

Stop words can have a significant impact on topic modeling and the performance of a classifier. With that in mind, stop words were removed from the corpus before conducting topic modeling and the train test split for classification, decreasing the total corpus size by -18.93%. The stop words list was obtained from the paper *Effects of Stops Words Elimination for Arabic Information Retrieval: A Comparative Study*, which examines and compares the effects of term weighting and stop words on Arabic retrieval. The paper found that the overall performance of the 1590-word General Stoplist generally performed better than other lists developed for the paper (El-Khair, 2017). We pass the resulting corpus through Gensim's Dictionary representation to conduct topic modeling. Additionally, we remove terms that occur too frequently (over 60%) or too rarely (less than 20 occurrences in total).

3.1 Latent Dirichlet allocation (LDA)

We use Gensim's doc2bow method to get a bag of words representation, and then initialize a Latent Dirichlet allocation to explore the linear combinations of the features that separate the two dialects. We interactively visualize 50 topic models that constitute each dialect by utilizing the LDAvis tool (LDA topic models available here).

The topics are further broken down with Principle Component Analysis (PCA) into 2 principle components, and plotted with a dendrogram representing 50 LDA topics.

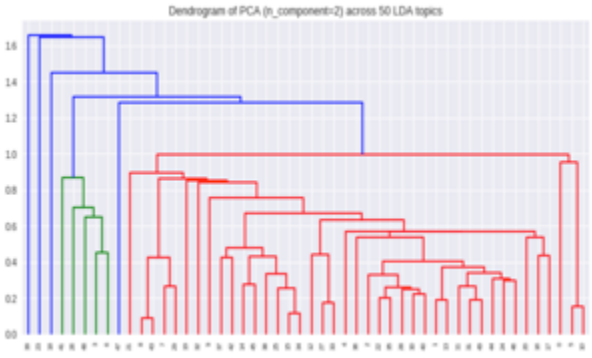

Figure 6: Topics appearing closely together are more likely to represent the same dialect.

3.2 Latent Semantic Analysis (LSA)

To further analyze dialectal relationships across the different tweets, we initialize a term frequency–inverse document frequency (TF-IDF) model using Gensim to transform the corpus for use with LSA. We then initialize an LSA transformation with 500 topics to find cosine similarities between the LSA topics and a query. Using a dialectal tweet belonging to the EG class as a test, we turn the tweet it into a bag of words to be used as a query with LSA and project it onto a 2-d vector space (topics), where 0 represents EG and 1 represents GLF.

" مليش دعوه بحد بصراحه ولا نادى ليه دعوه بصفقات حد السوق"

(0, 0.256), (1, -0.065)

We then perform a similarity query against the first 10 entries in the corpus, which have been annotated with the class EG.

```
[   (0, 0.97553933),
    (1, 0.979222),
    (2, 0.99163806),
    (3, 0.97607386),
    (4, 0.99301887),
    (5, 0.96669626),
    (6, 0.98321593),
    (7, 0.56111413),
    (8, 0.56111413),
    (9, 0.99779427)]
```

Figure 7: The first 10 documents (tweets) are labeled with class EG and the query itself is written in EG. The results show high similarity with most documents and two possible misclassifications (documents 7, 8).

4 Classification (EG & GLF)

We label encode the classes EG and GLF and use the General Stoplist (El-Khair, 2017) to remove stop words from the corpus. A train/test split is conducted and the value of the test is set to 0.25. The resulting text is run through a pipeline that includes a Count Vectorizer, TF-IDF Transformer, and a Multinomial Naive Bayes classifier with default (benchmark) sklearn hyperparameter settings.

4.1 Classification Report

A heatmap was created to illustrate the results of our Classification Report in Figure 8, showing the performance of our models' recall, precision, and f1-score (the harmonic mean of precision and recall), all of which are over 90%.

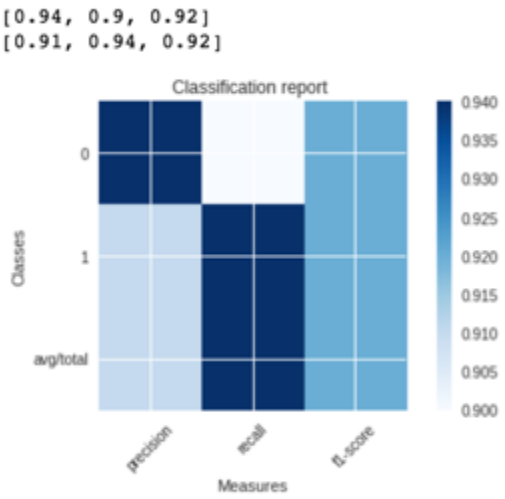

Figure 8: EG shown as 0 and GLF shown as 1. With 0.94 precision and 0.9 recall score for EG, and 0.91 precision and 0.94 recall for GLF. The F1 score for both classes is 0.92.

4.2 Confusion Matrix (raw values)

The raw values of the classification results are further explored through a confusion matrix to quantify the classifier's performance.

Predicted	0	1	All
True			
0	34328	3631	37959
1	2293	35483	37776
All	36621	39114	75735

Figure 9: A total of 75,735 tweets are used for the test set, with 34,328 predicted accurately for EG and 35,483 predicted accurately for GLF.

4.3 Confusion Matrix (percentages)

The raw values of the confusion matrix are converted to percentages in figure 10, showing that the classifier misclassified EG for GLF 4.64% of the time and GLF for EG only 3.13%.

Predicted	0	1	All
True			
0	46.87	4.64	25.06
1	3.13	45.36	24.94
All	50	50	50

Figure 10: Out of all EG tweets, 46.9% were correctly classified, while 45.4% of GLF tweets were correctly classified.

4.4 AUROC

To explore which models are best at predicting the classes, we plot the Area Under the Receiver Operating Characteristic curve (AUROC). This allows us to combine the True positive rate (TPR) and the False positive rate (FPR) into a single metric, an AUROC score of 0.97.

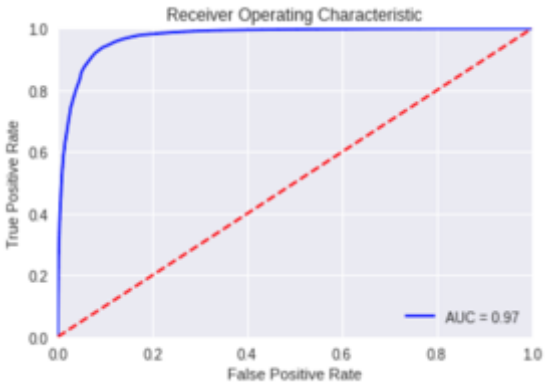

Figure 11: An optimal threshold can be achieved under a FPR of 0.2.

5 Conclusions and Future Work

Using a benchmark Naive Bayes on a test set of 75,735 manually annotated test tweets achieved a 92% dialect prediction score, with high AUC/ROC (0.97) and high f1 scores (.92 for both classes), suggesting that the classifier can do a good job for many other values of

threshold. These results show that dialect trained models can improve accuracy when compared to a system trained solely on an MSA-English parallel corpus.

Social media is a good source of dialectal data that can be used to train dialect identification classifiers. By refining models created for this paper, we intend to prototype dialect identifier tools that will be available online and incorporate a user feedback feature. As the corpus size was intended to build a proof of principle, we will continue expanding the corpus and adding new dialects and languages. Experimenting with different term weights, models, and stop words, as well as conducting deep learning text classification using word embeddings (Keras) can further improve results. Gensim also includes numerous features that were not utilized, including use word2vec, doc2vec, BM25. Tagging the words with The Stanford Arabic Parser, which includes a built-in Arabic Tree Bank (Maamouri, 2005) and Arabic Segmenter can also be used before term weighting to only keep nouns and verbs, which can further improve results.

References

Ibrahim Abu El-Khair. 2017. *Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study*. International Journal of Computing & Information Sciences 4 (3), 119-133. http://adsabs.harvard.edu/cgi-bin/bib_query?arXiv:1702.01925

Marco Bonzanini. 2016. *Mastering Social Media Mining with Python*. Packt Publishing. Birmingham, United Kingdom.

Mohamed Maamouri (project head). 2005. *Arabic Treebank*. Linguistic Data Consortium, PA.

Mona Diab and Nizar Habash. Natural Language Processing of Arabic and its Dialects. *In EMNLP 2014, Doha, Qatar.*

Mo3jam. 2017. User-generated dictionary of colloquial Arabic. https://ar.mo3jam.com/

Omar F. Zaidan and Chris Callison-Burch. 2014. *Arabic Dialect Identification.* Comp. Ling. vol. 40 issue 1, pp. 171-202.

Salem, F. (2017). The Arab Social Media Report 2017: Social Media and the Internet of Things: Towards Data-Driven Policymaking in the Arab World (Vol. 7). Dubai: MBR School of Government.

Steven Bird, Ewan Klein, Edward Loper. 2009. Natural Language Processing with Python. Analyzing Text with the Natural Language Toolkit. O'Reilly Media.

Zeinab Ibrahim. 2004. Polysemy in Arabic Dialects. In *Investigating Arabic Current Parameters in Analysis and Learning*, pages 51-64. Brill. https://www.academia.edu/13143254/_Polysemy_in_the_Arabic_ialects_