Identifying Distinguishing Features of African English Dialects

Anonymous ACL submission

Abstract

This paper documents the application of the CRISP-DM methodology to African English dialect sub-corpora and the extraction of distinguishing key features of the Ghanaian English dialect. The corpora were classified, and the models evaluated using the Waikato Environment for Knowledge Analysis (WEKA) tool after collecting the corpora with SketchEngine.

1 Introduction

Ghana has a population of 28 million people, and yet over 50% of its population speak English. Much like the USA-UK divide in the English language, one would think that such a divide would exist amongst dialects from African English-speaking countries, including Ghana, Zambia, Nigeria, Kenya, and Morocco.

2 Business Understanding

2.1 Objectives

The primary objective is to find features of the Ghanaian-English dialect that distinguish it from other African English dialects. The secondary objective is to experiment with machine learning classifiers to produce an accurate and precise model of the African-English dialects within WEKA, which can be helpful when trying to identify said features.

2.2 Requirements

Each corpus used to produce the models should be balanced (there should be a similar number of instances per nominal class) and it should be cleansed of the artefacts of data collection before modelling. Of course with a corpus limited to around 50,000 words, dimensions of data quality (generality, diversity, applicability, etc.) may not be satisfied fully. For example, diversity likely is not fully achievable in such a small corpus, but there is an expectation it is general – it covers a generous portion of the country's English dialect and applicable – there is a level of generality of the words collected to the dialect.

Good performance is also an important quality of the models to be produced. Models with 100% accuracy are preferable but this is not realistic. Any model with an accuracy above 70% is acceptable and will be used in the evaluation phase.

2.3 Problem Definition

The problem can be modelled as a question: Can corpora of African English dialects be cleansed, filtered, classified, and evaluated and can distinguishing features of the Ghanaian English dialect be identified? It is expected that there will be numerous models classifying the African English dialects, as well as a selection of distinguishing key features of Ghanaian English with respect to the other dialects.

3 Data Understanding

3.1 Data Format and Content

The raw corpora were exported from SketchEngine as text files (.txt), which contained artefacts of the SketchEngine corpus format, including markup-like tags such as <doc> and <p>. Each <doc> tag encapsulated a collection of <p> tags, where each <p> tag is a string of text (each word in said string being a word in the corpus). The <doc> tags correspond to the different documents downloaded within SketchEngine to construct the corpus. Additionally, each <doc> tag has a number of attributes to describe the location of the corresponding document (e.g. url, parent_folder, id, file_name).

The corpora themselves contained a wide variety of documents such as news and tutorial articles, various tabloid-like celebrity 'gossip' articles and some religious articles discussing religious issues in their respective countries.

3.2 Data Quality Issues

Although there is quite a wide variety of articles or sources in the corpora, there are quite a few words related to fashion and clothing, perhaps due to the 'summer' seed term I chose to use within SketchEngine. This degrades the generality of the dialect data-set but is expected with its limited size.

One prevalent issue with data quality was that the corpora contained domains and usernames from forums. Website domains and usernames are not necessarily features of the dialects, thus degrading data-set applicability or quality. A similar issue arises with citations or sources in the corpora. Examples of these sources include the likes of "ebay" and "pixabay.com". Another issue with the corpora was non-alphanumeric characters, which may skew the training accuracy and are not unique features of any of the dialects.

4 Data Preparation

4.1 Cleaning the Ghanaian Corpus

Before importing the Ghanaian English corpus to filter and classify within WEKA, it was cleansed. This was done manually using a text editor's find and replace tool to ensure that the markup tags, website domains, forum usernames and non-alphanumeric characters were removed before the conversion to the ARFF format, as seen in table 1.

Before	After
<doc url="https://yen.com.gh/107482-how-write-a-formal-letter-ghana.html" parent_folder="web1" id="file19525950" filename="107482-how-write-a-formal-letter-ghana.html">	Line 1 is removed as it is a markup tag and is not part of the corpus.
Ankara African wear shirts for men	Ankara African wear shirts for men
@fulungwana said:	said:

Table 1: Examples/Evidence of corpus cleaning.

4.2 Preparation and Further Cleaning

The next step was to compile the 5 corpora from Ghana, Zambia, Nigeria, Kenya, and Morocco into a single ARFF data-set with 2 attributes. The first attribute is a string attribute corresponding to a line of text from one of the corpora, and the second attribute is the nominal class attribute 'Country' describing which dialect corpus the line was from, which can be one of the countries listed above. The ARFF header is seen in figure 1.

```
@relation AfricanEnglish
@attribute "Document Text" string
@attribute Country
{Ghana,Zambia,Nigeria,Kenya,Morocco}
@data
```

Figure 1: AfricanEnglish data-set header.

There was an attempt by each team member to clean their respective corpus before sharing it, and each line from each corpus was placed into the AfricanEnglish data-set as seen in table 2.

Before	After
Attitude is a little thing that makes a big difference.	"Attitude is a little thing that makes a big difference.",Ghana
I thank you, Sir.	"I thank you, Sir.",Zambia

Table 2: Conversion to ARFF examples.

The other corpora in the data-set were not fully clean when the team sent them, so I did further cleaning before producing the ARFF data-set. The primary issue was removing more non-alphanumeric data, particularly prevalent in the Zambian and Nigerian corpora. This was a case of running a custom Python script to remove said data. If left untouched, this could have skewed the training accuracy as non-alphanumeric data is not representative of the dialects but are artefacts of the online documents downloaded into SketchEngine. The final data-set contained 1947 Ghanaian instances, 3590 Zambian instances, 1377 Nigerian instances, 2666 Kenyan instances and 2723 Moroccan instances. This was not a well-balanced data-set, but this was expected with the size constraints of each sub-corpus.

5 Modelling

5.1 Experimenting with Classifiers

Alshutayri, E. Atwell, A. Alosaimy, J. Dickins, M. Ingleby, and J. Watson. (2016) demonstrated the use of the classifiers ZeroR, NaiveBayes, SMO and J48, which I chose to experiment with due to their variety of training accuracies. A. Alshutayri and E. Atwell. (2017) also demonstrated the use of a "Multinomial Naive Bayes (MNB) algorithm with

WordTokenizer", which inspired me to explore the NaiveBayesMultinomialText classifier, a Multinomial Naive Bayes algorithm that operates directly on text data. Each classifier (excluding NaiveBayesMultinomialText) was applied to the data-set using the meta classifier FilteredClassifier, which allowed for the StringToWordVector to be used to convert string features into nominal words, which are subsequently converted into numeric word-occurrence features. NaiveBayesMultinomialText was applied directly without the need for the StringToWordVector filter as it is designed to operate directly on string attributes, but still classifies the numeric word-occurrence information internally.

Classifier	Training Set	Percentage Split (60% training-40% testing)	10-Fold Cross Validation
ZeroR	29.18%	29.02%	29.18%
NaiveBayes	52.74%	52.08%	51.95%
J48	78.31%	62.06%	65.74%
SMO	90.60%	72.89%	75.59%
NaiveBayes MultinomialText	88.84%	78.40%	80.71%

Table 3: Accuracy of different classifiers.

As seen in table 3, NaiveBayesMultinomialText emerged the best performing out of all the classifiers when using default settings, with the SMO classifier as a close second, and the J48 decision tree classifier as a close third. The best classifiers to evaluate would likely be any with an accuracy above the baseline 70% as specified in the requirements, namely SMO and NaiveBayesMultinomialText.

When actually evaluating the models, 10-Fold Cross Validation is the better testing method, not because of its accuracy, but because it avoids testing on just one configuration of the data-set (which is what Percentage Split does). Testing on the training set should also be avoided during evaluation, even though its accuracy is appealing, as I want to avoid overfitting.

5.2 Features and Parameter Settings

By default, NaiveBayesMultinomialText and StringToWordVector use WordTokenizer to produce occurrence information for individual words from string features in the data-set, meaning each word in the data-set becomes a feature. Interestingly, WEKA has additional tokenizers such as NGramTokenizer which can be used to tokenize string features differently.

NGramTokenizer, for example, divides strings into n-grams rather than words, which could be used to discover interesting dialect-unique combinations of words that could have dialect-specific contexts. This tokenizer of course is more computationally expensive, and also generally yields a slightly lesser accuracy when used, as seen in table 4.

Classifier	10-Fold Cross Validation
ZeroR	29.18%
NaiveBayes	52.27%
J48	65.74%
SMO	73.88%
NaiveBayesMultinomialText	77.92%

Table 4: Accuracies using NGramTokenizer.

I experimented with NaiveBayesMultinomialText and achieved 80.87% accuracy by changing the 'lowercaseTokens' parameter to true. Adjusting other settings seemed to adversely affect training accuracy.

```
   a    b    c    d    e   <-- classified as
1703   74   86   24   60 |    a = Ghana
 115 3076   65  107  227 |    b = Zambia
 122   79 1066   29   81 |    c = Nigeria
 140  250   74 2069  133 |    d = Kenya
 145  349  103   90 2036 |    e = Morocco
```

Figure 2: Confusion matrix for model 1.

This will be evaluated as Model 1 (see figure 2), but Model 2 (see figure 3) will use NGramTokenizer instead of WordTokenizer. Model 2's 'lowercaseTokens' was also set to false as when true, it significantly reduced training accuracy.

```
   a    b    c    d    e   <-- classified as
1563   48  272   12   52 |    a = Ghana
 105 2983  168   82  252 |    b = Zambia
  61   77 1181   16   42 |    c = Nigeria
  88  266  310 1855  147 |    d = Kenya
 100  266  301   51 2005 |    e = Morocco
```

Figure 3: Confusion matrix for model 2.

I also experimented with SMO by changing the complexity parameter exponentially from 0.001 to 100.0 until I discovered the optimal value, which happened to be the default value: 1.0. I ended up using the default linear PolyKernel rather than the gaussian RBFKernel as I could not find appropriate gamma/complexity parameter values to yield an acceptable training accuracy.

```
    a    b    c    d    e   <-- classified as
 1476  260   50   38  123 |    a = Ghana
   65 3381   34   24   86 |    b = Zambia
  127  250  853   40  107 |    c = Nigeria
   66  759   37 1692  112 |    d = Kenya
  105  967   61   45 1545 |    e = Morocco
```

Figure 4: Confusion matrix for model 3.

This will be evaluated as Model 3 (see figure 4) and Model 4 will use NGramTokenizer (see figure 5).

```
    a    b    c    d    e   <-- classified as
 1555  164   79   46  103 |    a = Ghana
   78 3275   40   53  144 |    b = Zambia
  174  196  871   44   92 |    c = Nigeria
   91  514   62 1880  119 |    d = Kenya
  134  679   94   97 1719 |    e = Morocco
```

Figure 5: Confusion matrix for model 4.

6 Evaluation

6.1 Evaluation Methods

When evaluating a model, it is typical to evaluate its performance using a number of outputs from model testing. For example, the performance of the model can be assessed by examining the accuracy, precision and recall of each classifier and comparing the model with others using WEKA.

6.2 Results

Figures 2-5 show the confusion matrices for models 1-4 respectively. By observation, Model 1 has the fewest false negatives when looking at the Ghana class with only 244 incorrectly classified instances, whereas Model 3 had the most with 392 instances being incorrectly classified. The general case (considering all classes) is slightly different (as seen in table 5), with Model 1 yet again showing the best precision and recall and thus the greatest relevancy. Model 4 however, shows the worst precision and recall and thus has the poorest performance.

Measure	Model 1	Model 2	Model 3	Model 4
accuracy	0.809	0.779	0.756	0.739
precision	0.812	0.804	0.771	0.756
recall	0.809	0.779	0.756	0.739

Table 5: Measures for models 1-4 (weighted avg.).

The models were then compared in WEKA's experimenter using their F-measure values (seen in figure 6) to better understand each classifier's performance. Model 1's performance was pretty much indistinguishable to Model 2. The asterisk beside the F-measure values for Models 3 and 4 suggest they performed significantly worse than Models 1 and 2.

```
 (1) bayes.N | (2) baye (3) meta (4) meta
-----------------------------------------
(10)   0.82 |   0.81     0.78 *   0.77 *
-----------------------------------------
      (v/ /*) |  (0/1/0)  (0/0/1)  (0/0/1)
```

Figure 6: F-measure comparison of Models 1-4.

6.3 Best Features and Classifiers

E. Atwell, J. Arshad, C. Lai, L. Nim, N. Rezapour Ashregi, J. Wang, and J. Washtell. (2007) showed a "lexical level" of analysis that involved counting word occurrence information to compare spellings of words in UK and USA English sub-corpora, and T. Tarmom, W. Teahan, E. Atwell, and M.A. Alsalka. (2020) used the "top 10 most frequent words" as a metric of difference between sub-corpora. This helped me develop a method to 'extract' key features: I examined the word frequencies of features in the Ghana class within the data-set and used the top 10 most 'unique' features to find the most distinguishing of them all. This involved some research into the word's meanings and revisiting SketchEngine to examine key words/multi-words. A lot of features examined were either names of places, people, or regional terminology (seen in table 6).

Word Feature	Justification
kpone-katamanso	It is one of the governmental constituencies in Ghana.
own Kente style	Kente refers to a Ghanaian textile.
Kubekro	Kubreko river in Ghana.
Kente cloth	Kente refers to a Ghanaian textile.

Table 6: Best features.

Table 5 and figure 6 demonstrated that Model 1 had the best training accuracy, precision, recall and thus f-measure of the four classifiers. Model 2's performance was promising too, with only marginally smaller precision and recall. These were the best models, both using the NaiveBayesMultinomialText classifier.

7 Conclusion

Through the CRISP-DM methodology, my team and I collected a number of African English sub-corpora to independently prepare, model, and evaluate. After showing I could model the dialect data-set, I identified distinguishing features of the Ghanaian English dialect.

References

Alshutayri, E. Atwell, A. Alosaimy, J. Dickins, M. Ingleby, and J. Watson. 2016. Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts. In *Proceedings of VarDial'2016 Workshop on NLP for Similar Languages, Varieties and Dialects*, pp. 204-211

E. Atwell, J. Arshad, C. Lai, L. Nim, N. Rezapour Ashregi, J. Wang, and J. Washtell. 2007. Which English dominates the world wide web, British or American? *Proceedings of CL'2007 Corpus Linguistics Conference.*

T. Tarmom, W. Teahan, E. Atwell, and M.A. Alsalka. 2020. Compression versus traditional machine learning classifiers to detect code-switching in varieties and dialects: Arabic as a case study. *Natural Language Engineering Journal.*

A. Alshutayri and E. Atwell. 2017. Exploring Twitter as a Source of an Arabic Dialect Corpus. *International Journal of Computational Linguistics (IJCL).* 8(2), pp. 37-44.