Feature Identification of Spanish Dialects

Anonymous ACL submission

Abstract

We present the results of a binary classification project undertaken in WEKA on the Spanish language as used in the Philippines. Notable is the high 98.4% classification accuracy when applying the Sequential Minimal Optimisation algorithm (Platt, 1998; Keerthi et al., 2001; Hastie and Tibshirani, 1998) to attributes obtained as character N-gram strings encoded as vectors using the in-built "StringToWordVector" WEKA filter. Sketch Engine is implemented for automated web corpus collection.

1 Introduction

This paper will attempt to identify language features that distinguish various Spanish dialects from different countries. Specifically, Spanish-speaking countries considered herein are: Chile, Colombia, Cuba, Mexico, Peru, and the Philippines. We will attempt to build a classifier within WEKA that is able to distinguish between these dialects, and hence identify the features with which it is able to do so. This paper will follow the CRISP-DM data mining methodology and will refer to the phases in the order they were considered during the proceeding of this project. Though there is no stakeholder for this project, we will outline the goals and criteria to which we can refer for a quantification of success within our evaluation and presentation phases.

2 Business Understanding

We present the following objectives that we expect to accomplish for this project:

1. Collate 50,000 words for each specified national dialect;
2. Parse files into a format suitable for data analysis;
3. Create a binary classifier in WEKA to classify between Spanish from the Philippines and Spanish from another national dialect;
4. Identify features used by the classifier to determine its classification;
5. Consider the potential applications of the results of this project.

We will address these objectives in the subsequent phases of the CRISP-DM methodological process. An evaluation of the fulfilment of these objectives will be presented in the appropriate phase. We now present a brief summary of how the parameters of these goals were selected.

A target of 50,000 words for each dialect was selected as a large enough amount from which a classifier could be trained, whilst maintaining a relatively manipulable data set of size 300,000 words. WEKA was chosen to create a model due to ease of use and range of readily available classifiers, drastically reducing the time to complete the project. A binary classifier was designated as a starting point for the project, which could be extended upon based on the results found.

The decision to focus on Spanish in the Philippines was taken with the knowledge of Spanish colonial history and the current remnants of the language, being designated as a protected and official language. Indeed there are some differences in linguistic features, for example spelling (Dimaculangan, 2017), that may be used for the classification of Spanish sentences.

3 Data Understanding

Collation of corpora was done through the proprietary software Sketch Engine - chosen for its

UTF-8 encoding of web corpora (Atwell and Alfaifi, 2016), allowing Spanish accented characters - using its web scraper WebBootCaT to scrape text from Spanish web pages hosted in the relevant country (via top level domain restriction). The five seed terms "Español," "clima," "comida," "mar," and "iglesia" were passed to WebBootCaT to facilitate the search of web documents with these subjects - kept intentionally neutral to avoid niche subject areas and ensure a large volume of data. From this process, we obtained six corpora as ".txt" files, listing a sequence of HTML documents followed by lines of text taken from these documents.

Some brief inspection of the data was undertaken to verify the integrity of the text, for example checking the text was indeed Spanish, checking for formatting errors, and ensuring the text was properly encoded in UTF-8 format. This was performed to assure smoothness in the data cleaning process described in section 4. The remainder of the data understanding was completed after cleaning and importing the text into WEKA.

Visual inspection of the text files revealed the inclusion of figures, special characters (for example "@" and "#"), and large areas of whitespace which would all require removal from the files. Hence, this activity was useful in informing exactly what data cleaning needed to take place when parsing the text into a machine-readable format.

4 Data Preparation

Importing data into WEKA requires a format such as CSV or ARFF. For simplicity, a CSV format was chosen, which was then converted by WEKA into ARFF format for ease when dealing with string data within the program. Python was chosen for parsing the text corpora files into a format to be read into WEKA due to its ease of reading from and writing to files.

The code snippet in Figure 1 demonstrates the core functionality of the Python parser, using string methods and regular expressions to clean individual lines. Lines 4 to 8 remove the "<doc>' and "<p>" HTML tags left in the text by Sketch Engine. Line 11 removes special characters not contained within the modern English alphabet and preserves the characters used in Spanish not contained within the English alphabet, such as "á" and "é". All lines of text are then converted to lower case, punctuation is stripped, and all empty lines are removed. This was done to aid comparison of vocabulary, spelling, and word order. However, if differences in punctuation and sentence structure should be considered, then punctuation would not need to be stripped. Since we have opted for this removal, our model will not account for punctuation when classifying sentences as Spanish from the Philippines or not.

Average Merit	Average Rank	Attribute
0.028 ± 0.001	1 ± 0	1429 paul
0.021 ± 0.001	2.4 ± 0.49	758 que
0.021 ± 0.001	2.6 ± 0.49	1128 cristina
0.019 ± 0.001	4 ± 0	488 le
0.017 ± 0.001	5.8 ± 0.6	1336 malik
0.016 ± 0.001	5.8 ± 0.98	411 había
0.016 ± 0.001	6.7 ± 1.1	608 no
0.014 ± 0	9.4 ± 1.02	323 estaba
0.014 ± 0.001	9.7 ± 1.95	681 pero
0.014 ± 0.001	9.7 ± 1.62	764 qué

Table 1: Top ten attributes returned from attribute selection

The lines generated by this parsing process are then written to a file in a CSV format with two attributes: one for the line and one designating the class of that line (either Philippines or not Philippines). Hence, we have generated a file in CSV format where each sentence extracted from Spanish web pages is classed as either being from a website in the Philippines or not.

5 Modeling

We may now import our data into WEKA and construct a model for dialect classification. Since we are considering strings, we must apply the "StringToWordVector" filter to perform analysis and obtain information from our data set; opting for a minimum term frequency of three, hence removing anomalous terms from the data. WEKA's default "WordTokenizer" was used as the tokenizer for this filter.

Continuing our exploration of the data set, we perform attribute selection with the preset "InfoGainAttributeEval" evaluator in WEKA, specifying a threshold of 0. That is, we only select the attributes that give us information about how a classification could be derived. Selecting with 10-fold cross validation, we obtain the top ten results shown in Table 1. Three of these attributes (1429, 1128, and 1336) appear to be names used fre-

```
#Parse lines for writing
for line in lines:
    #'if' condition removes all lines with <doc> tags
    if '<doc ' not in line and '</doc>' not in line:

        #Remove <p> tags
        parsed_line = line.replace('<p> ', '')
        parsed_line = parsed_line.replace('</p>', '')

        #Remove non-letters and convert to lower case
        parsed_line = re.sub('[^A-Za-z\áéíñóúüÁÉÍÑÓÊÚÜs]+', '',
            parsed_line).lower()[:-1]

        #Get rid of empty lines
        if parsed_line != '' and parsed_line != '\n':
            parsed_lines.append(parsed_line)
```

Figure 1: Parser Snippet

quently in one or more documents. Since we want a classifier to consider general language use, not the appearance of specific names in documents, we remove these points from the data set before applying a model to the data. Indeed visual inspection of the corpus collected from websites in the Philippines shows a single document containing frequent use of the names “Paul” and “Cristina” within some form of letter or message.

With our new data set we may now apply the ZeroR classifier to obtain a baseline accuracy, upon which we may judge the performance of our other WEKA classifiers. However, since we are currently working with a class imbalance (1755 from the Philippines and 13106 not), we must first apply a second filter to our data to remove any class bias when classifying the data, specifically during the training. Applying the WEKA “Resample” filter with a uniform bias parameter of 1.0 gives us a new class balance of 7430 lines for both classes. Applying ZeroR with this filter gives the expected accuracy of 50% to which we may now compare the accuracy of other classifiers.

The following classifiers were used on this data set using 10-fold cross validation: Naive Bayes; Bayesian Network; SMO; J48. The results are summarised in Table 2. We omit the recall since it is equal to the true positive rate, as we do not consider positive and negative classes. Evidence for these results are placed in Appendix A

Subsequently, this process was repeated but using “CharacterNGramTokenizer” and “NGramTokenizer” as the tokenizer for the “StringToWordVector” filter, with a maximum N-gram size of 10. This was done in an attempt to find more complex language features in the form of word N-grams and character N-grams (Alshutayri et al., 2016). The results of attribute selection are presented in Table 3 and Table 4. Classification results are provided in Table 5 and Table 6 and evidenced in Appendix B and Appendix C.

Classifier	TP rate	FP rate	Prec.	Build Time (S)
NaiveBayes	0.719	0.281	0.722	2.97
BayesNet	0.801	0.199	0.817	6.72
SMO	0.923	0.077	0.925	118.04
J48	0.912	0.088	0.913	721.76

Table 2: Classifier results

6 Evaluation and Deployment

As was expected due to its high performance in other studies (Alshutayri et al., 2016; Tarmom et al., 2020), the SMO classifier had the highest individual accuracy of 98.4% when combined with the “CharacterNGramTokenizer” tokenizer in the “StringToWordVector” filter. The average accuracy for each classifier across all three tests was 69.3%, 74.5%, 93.7%, and 92% for Naive Bayes, Bayesian Network, SMO, and J48 respectively. Hence we see that the probabilistic classifiers consistently under-performed compared to SMO and J48. However, this came at the cost of greatly in-

Average Merit	Average Rank	Attribute
0.059 ± 0.001	1 ± 0	124 _qu
0.058 ± 0.001	2 ± 0	123 _q
0.051 ± 0.001	3 ± 0	125 _que
0.048 ± 0.001	4.3 ± 0.64	786 qu
0.048 ± 0.001	5.5 ± 0.81	785 q
0.047 ± 0.001	6.3 ± 1.62	1055 aba
0.047 ± 0.001	6.9 ± 0.83	787 que
0.046 ± 0.001	7.8 ± 0.87	126 _que_
0.046 ± 0.001	8.5 ± 1.63	72 _ha
0.045 ± 0.001	10.4 ± 0.8	206 ab

Table 3: Attribute selection results with "CharacterNGramTokenizer"

Average Merit	Average Rank	Attribute
0.046 ± 0.001	1 ± 0	689 que
0.036 ± 0.001	2.4 ± 0.49	542 no
0.035 ± 0.001	2.6 ± 0.49	448 le
0.03 ± 0.001	4.4 ± 0.49	1241 estaba
0.029 ± 0.001	5.3 ± 0.9	454 lo
0.029 ± 0.001	5.4 ± 0.8	641 pero
0.027 ± 0.001	7 ± 0.45	1279 había
0.025 ± 0	8.4 ± 0.49	258 era
0.025 ± 0.001	8.5 ± 0.67	471 me
0.021 ± 0.001	10 ± 0	715 qué

Table 4: Attribute selection results with "NGramTokenizer"

Classifier	TP rate	FP rate	Prec.	Build Time (S)
NaiveBayes	0.646	0.354	0.654	3.66
BayesNet	0.630	0.370	0.633	9.88
SMO	0.984	0.016	0.985	1201.45
J48	0.935	0.065	0.937	160.14

Table 5: Classifier results with "CharacterNGramTokenizer"

Classifier	TP rate	FP rate	Prec.	Build Time (S)
NaiveBayes	0.715	0.285	0.717	3.33
BayesNet	0.804	0.196	0.804	8.37
SMO	0.904	0.096	0.906	147.31
J48	0.913	0.087	0.913	738.03

Table 6: Classifier results with "NGramTokenizer"

creased build times - up to over 120 times the build time in the case of applying "CharacterNGramTokenizer." The relatively comparable results of the default "WordTokenizer" and "NGramTokenizer" suggests that the features of the Spanish Philippines dialect lies within the character N-grams. This conclusion is reinforced by the similarity in the attribute selection results in Table 1 and Table 4.

Table 3 suggests that the placement of word fragments such as "que" and "qu" can be used to differentiate Spanish from the Philippines and Spanish from the other selected dialects, as can be seen by the placement of whitespace within the attributes, indicated by an underscore. Clearly the presence of a classifier result with almost 100% accuracy demonstrates that there is indeed some difference within the character N-grams of Spanish from the Philippines and Spanish from elsewhere in the world.

In terms of business uses of these results and deployment of the classifiers, certainly these results may be helpful in the improvement of translation services. In particular, a classifier of this kind could be used to determine the dialects of phrases to be learned by a machine translation system, with the goal of developing more geographically local translations in terms of language use and sentence structure.

Considering the objectives set for the project at the beginning of this paper, it is the belief of the author that this project has been successful and has fulfilled all necessary criteria. The use of Sketch Engine went smoothly and was a much better suited alternative to manual collation of corpora (Al-sulaiti et al., 2016), providing easy-to-parse text files for analysing in WEKA. SMO proved to be a powerful classification method and some features of the dialect were identified in the form of character N-grams. This project could be easily extended to classify between each of the six aforementioned dialects, however it is likely that a larger data set would be needed, as shown when re-sampling was required for binary classification.

Acknowledgments

Many thanks to the anonymous five team members of the author - who collected the Spanish dialect corpora for Chile, Colombia, Cuba, Mexico, and Peru - for their assistance and input with corpus collection.

References

Latifa Al-sulaiti, Noorhan Abbas, C Brierley, Eric Atwell, and Ayman Alghamdi. 2016. Compilation of an arabic children's corpus. In *Proceedings of LREC'2016 Language Resources and Evaluation Conference*.

Areej 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*, pages 204–211.

E. Atwell and A. Alfaifi. 2016. Comparative evaluation of tools for arabic corpora search and analysis. *International Journal of Speech Technology*, 19(2):347–357.

Shelly C. Dimaculangan. 2017. Filipino and spanish words: Spelling the difference.

Trevor Hastie and Robert Tibshirani. 1998. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press.

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. 2001. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649.

J. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Taghreed Tarmom, William Teahan, Eric Atwell, and Mohammad Ammar Alsalka. 2020. Compression versus traditional machine learning classifiers to detect code-switching in varieties and dialects: Arabic as a case study. *Natural Language Engineering*, 26(6):663–676.

A Classifier Results

```
Time taken to build model: 2.97 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10682               71.8843 %
Incorrectly Classified Instances      4178               28.1157 %
Kappa statistic                          0.4377
Mean absolute error                      0.2879
Root mean squared error                  0.4921
Relative absolute error                 57.5734 %
Root relative squared error             98.4143 %
Total Number of Instances            14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.782    0.345    0.694      0.782    0.736      0.441    0.795     0.797     not_philippines
                 0.655    0.218    0.751      0.655    0.700      0.441    0.795     0.789     philippines
Weighted Avg.    0.719    0.281    0.722      0.719    0.718      0.441    0.795     0.793

=== Confusion Matrix ===

    a    b   <-- classified as
 5812 1618 |    a = not_philippines
 2560 4870 |    b = philippines
```

Figure 2: NaiveBayes result

```
Time taken to build model: 6.72 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       11902               80.0942 %
Incorrectly Classified Instances      2958               19.9058 %
Kappa statistic                          0.6019
Mean absolute error                      0.1988
Root mean squared error                  0.4144
Relative absolute error                 39.7683 %
Root relative squared error             82.8889 %
Total Number of Instances            14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.914    0.312    0.745      0.914    0.821      0.618    0.909     0.915     not_philippines
                 0.688    0.086    0.889      0.688    0.776      0.618    0.909     0.906     philippines
Weighted Avg.    0.801    0.199    0.817      0.801    0.798      0.618    0.909     0.911

=== Confusion Matrix ===

    a    b   <-- classified as
 6791  639 |    a = not_philippines
 2319 5111 |    b = philippines
```

Figure 3: BayesNet result

```
Time taken to build model: 118.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       13720               92.3284 %
Incorrectly Classified Instances      1140                7.6716 %
Kappa statistic                          0.8466
Mean absolute error                      0.0767
Root mean squared error                  0.277
Relative absolute error                 15.3432 %
Root relative squared error             55.3953 %
Total Number of Instances            14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.953    0.107    0.899      0.953    0.926      0.848    0.923     0.881     not_philippines
                 0.893    0.047    0.950      0.893    0.921      0.848    0.923     0.902     philippines
Weighted Avg.    0.923    0.077    0.925      0.923    0.923      0.848    0.923     0.891

=== Confusion Matrix ===

    a    b   <-- classified as
 7084  346 |    a = not_philippines
  794 6636 |    b = philippines
```

Figure 4: SMO result

```
Time taken to build model: 721.76 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       13558               91.2382 %
Incorrectly Classified Instances      1302                8.7618 %
Kappa statistic                          0.8248
Mean absolute error                      0.1127
Root mean squared error                  0.2656
Relative absolute error                 22.5361 %
Root relative squared error             53.1228 %
Total Number of Instances            14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.933    0.109    0.896      0.933    0.914      0.825    0.960     0.960     not_philippines
                 0.891    0.067    0.930      0.891    0.911      0.825    0.960     0.947     philippines
Weighted Avg.    0.912    0.088    0.913      0.912    0.912      0.825    0.960     0.954

=== Confusion Matrix ===

    a    b   <-- classified as
 6935  495 |    a = not_philippines
  807 6623 |    b = philippines
```

Figure 5: J48 result

B Classifier Results with "CharacterNGramTokenizer"

```
Time taken to build model: 3.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       9602               64.6164 %
Incorrectly Classified Instances     5258               35.3836 %
Kappa statistic                         0.2923
Mean absolute error                     0.3536
Root mean squared error                 0.5919
Relative absolute error                70.7127 %
Root relative squared error           118.3701 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.759    0.467    0.619      0.759    0.682      0.300    0.690     0.638     not_philippines
                 0.533    0.241    0.689      0.533    0.601      0.300    0.699     0.675     philippines
Weighted Avg.    0.646    0.354    0.654      0.646    0.642      0.300    0.695     0.656

=== Confusion Matrix ===

    a    b   <-- classified as
 5639 1791 |    a = not_philippines
 3467 3963 |    b = philippines
```

Figure 6: NaiveBayes result

```
Time taken to build model: 9.88 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       9355               62.9542 %
Incorrectly Classified Instances     5505               37.0458 %
Kappa statistic                         0.2591
Mean absolute error                     0.3708
Root mean squared error                 0.6049
Relative absolute error                74.1555 %
Root relative squared error           120.9804 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.716    0.457    0.611      0.716    0.659      0.263    0.678     0.633     not_philippines
                 0.543    0.284    0.656      0.543    0.595      0.263    0.678     0.657     philippines
Weighted Avg.    0.630    0.370    0.633      0.630    0.627      0.263    0.678     0.645

=== Confusion Matrix ===

    a    b   <-- classified as
 5317 2113 |    a = not_philippines
 3392 4038 |    b = philippines
```

Figure 7: BayesNet result

```
Time taken to build model: 1201.45 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      14623               98.4051 %
Incorrectly Classified Instances      237                1.5949 %
Kappa statistic                         0.9681
Mean absolute error                     0.0159
Root mean squared error                 0.1263
Relative absolute error                 3.1898 %
Root relative squared error            25.2578 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.031    0.969      1.000    0.984      0.969    0.984     0.969     not_philippines
                 0.969    0.000    1.000      0.969    0.984      0.969    0.984     0.984     philippines
Weighted Avg.    0.984    0.016    0.985      0.984    0.984      0.969    0.984     0.977

=== Confusion Matrix ===

    a    b   <-- classified as
 7427    3 |    a = not_philippines
  234 7196 |    b = philippines
```

Figure 8: SMO result

```
Time taken to build model: 160.14 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      13901               93.5464 %
Incorrectly Classified Instances      959                6.4536 %
Kappa statistic                         0.8709
Mean absolute error                     0.0737
Root mean squared error                 0.2465
Relative absolute error                14.7455 %
Root relative squared error            49.2974 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.902    0.031    0.967      0.902    0.933      0.873    0.949     0.959     not_philippines
                 0.969    0.098    0.908      0.969    0.938      0.873    0.949     0.909     philippines
Weighted Avg.    0.935    0.065    0.937      0.935    0.935      0.873    0.949     0.934

=== Confusion Matrix ===

    a    b   <-- classified as
 6699  731 |    a = not_philippines
  228 7202 |    b = philippines
```

Figure 9: J48 result

C Classifier Results with "NGramTokenizer"

```
Time taken to build model: 3.33 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      10628               71.5209 %
Incorrectly Classified Instances     4232               28.4791 %
Kappa statistic                         0.4304
Mean absolute error                     0.2908
Root mean squared error                 0.4987
Relative absolute error                58.153  %
Root relative squared error            99.7484 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.764    0.333    0.696      0.764    0.728      0.432    0.794     0.804     not_philippines
                 0.667    0.236    0.738      0.667    0.701      0.432    0.794     0.784     philippines
Weighted Avg.    0.715    0.285    0.717      0.715    0.715      0.432    0.794     0.794

=== Confusion Matrix ===

    a    b   <-- classified as
 5674 1756 |    a = not_philippines
 2476 4954 |    b = philippines
```

Figure 10: NaiveBayes result

```
Time taken to build model: 8.37 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      11941               80.3567 %
Incorrectly Classified Instances     2919               19.6433 %
Kappa statistic                         0.6071
Mean absolute error                     0.2107
Root mean squared error                 0.4058
Relative absolute error                42.1446 %
Root relative squared error            81.1599 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.786    0.179    0.815      0.786    0.800      0.608    0.892     0.896     not_philippines
                 0.821    0.214    0.793      0.821    0.807      0.608    0.892     0.891     philippines
Weighted Avg.    0.804    0.196    0.804      0.804    0.804      0.608    0.892     0.894

=== Confusion Matrix ===

    a    b   <-- classified as
 5838 1592 |    a = not_philippines
 1327 6103 |    b = philippines
```

Figure 11: BayesNet result

```
Time taken to build model: 147.31 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      13436               90.4172 %
Incorrectly Classified Instances     1424                9.5828 %
Kappa statistic                         0.8083
Mean absolute error                     0.0958
Root mean squared error                 0.3096
Relative absolute error                19.1655 %
Root relative squared error            61.9121 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.132    0.877      0.940    0.908      0.810    0.904     0.854     not_philippines
                 0.868    0.060    0.936      0.868    0.901      0.810    0.904     0.878     philippines
Weighted Avg.    0.904    0.096    0.906      0.904    0.904      0.810    0.904     0.866

=== Confusion Matrix ===

    a    b   <-- classified as
 6986  444 |    a = not_philippines
  980 6450 |    b = philippines
```

Figure 12: SMO result

```
Time taken to build model: 738.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      13563               91.2719 %
Incorrectly Classified Instances     1297                8.7281 %
Kappa statistic                         0.8254
Mean absolute error                     0.1146
Root mean squared error                 0.2642
Relative absolute error                22.9112 %
Root relative squared error            52.8335 %
Total Number of Instances           14860

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.930    0.105    0.899      0.930    0.914      0.826    0.961     0.963     not_philippines
                 0.895    0.070    0.928      0.895    0.911      0.826    0.961     0.947     philippines
Weighted Avg.    0.913    0.087    0.913      0.913    0.913      0.826    0.961     0.955

=== Confusion Matrix ===

    a    b   <-- classified as
 6911  519 |    a = not_philippines
  778 6652 |    b = philippines
```

Figure 13: J48 result