Classification for National Dialect Corpus

University of Leeds
School of Mathematics
Leeds, UK

Abstract

The process of data mining helps to find many valuable patterns from a large amount of data. In this following paper, we will go through the advantage of using the common process model of data mining, such as the Cross-Industry Standard Process model (CRISP-DM). The entire experiment aims to compare the dialect of the single language with few other dialects of the same language. Also, to create and identify the classifier model. The elicitation of ideas is made possible by machine learning tools and techniques. Tools such as Sketch Engine and WEKA are used to aid in the learning process. This paper is focused on the overall pattern of all phases by systematically dissecting each phase.

1 Introduction

Data mining means getting valuable information from raw data. These days, due to the development of information technology, a large number of unstructured raw data like video, audio, text etc., are produced each day. Using machine learning classifiers, we can get useful structure information from unstructured data. In this experiment, I have worked with a group of 5 people to complete this data mining research. There are many languages in the world. Our task is to gather a text corpus of 50,000 words of the specific dialect of the same national language. Even though people speak the same language in a different country, pronunciation and text differ from country to country. We have chosen Spanish as our research language. The majority of people in many countries speak Spanish. So, each groupmate has collected 50,000 words from different dialects of different Top-Level-Domain (TLD). Text in the accumulated corpus is unstructured data. By the data mining process, collected data will be cleaned and changed into structured data. Then cleaned data will be stored in a dictionary. The main outcome of this experiment is to create a classifier. Basics of the text, classifier helps to find which dialect it belongs to. Stander Cross-Industry-Standard-Process (CRISP-DM) methodology helps to achieve the result, which includes Business Understanding and Data Understanding, and how we started preparing data and created a classifier model. The CRISP-DM methodology specifies a series of phases or sub-tasks in a data-mining project; it is a “recipe” to follow, allowing novices and non-experts to carry out data mining experiments successfully (Atwell, E et al., 2007). Many tools are used for searching and analysing corpora. They generally provide some basic functions (e.g., frequent words and concordances), whereas some of these tools have more functions and statistics such as collocations, n-gram/clusters, keywords, etc. (Alfaifi, A et al., 2016).

2 Business understanding

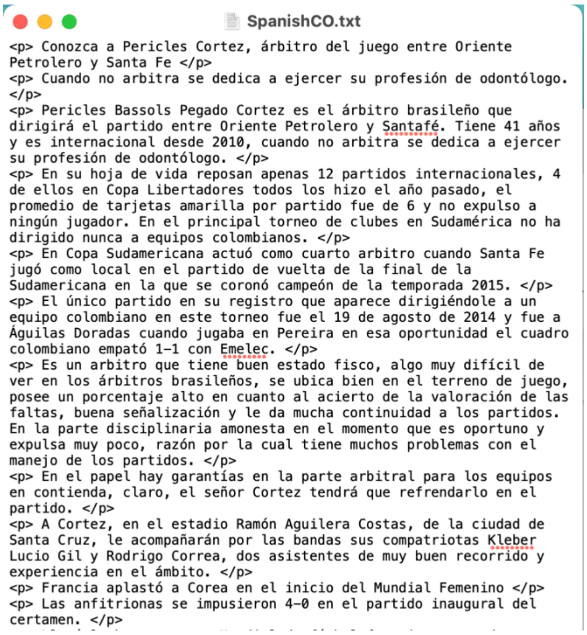
The cross-Industry standard process for data mining (CRISP-DM) is the dominant process framework for data mining. Even though we are a small group, we able to manage to gather the needed information for this experiment. Building a classifier with the help of some algorithm in WEKA or some other machine learning methodology. Due to a large number of datasets, it isn't easy to compute, and it consumes more time for processing the data. It requires more system power. It also uses the Waikato Environment for Knowledge Analysis (Weka) data analytic tool as a second method for the automatic detection of code-switching in text (Tarmom, T et al., 2020). Many collections of machine learning algorithm for data mining are available in WEKA, which is opensource software. The algorithm in it can be applied directly to the datasets or can be called from the program code. The collection of text corpus phase is done using the tool called Sketch Engine. In the upcoming part, we are going to create and train a model to classify the dialects. This analysis led us to conclude that it is impossible in principle for WEKA to classify all instances correctly (Alshutayri, A et al., 2016).

3 Data Understanding

Tools are very much essential to collect a corpus. There are many tools like WebCorp, Sketch Engine, Intellitext, etc., for collecting and analysing the text corpus. Here I have used Sketch Engine to collect my corpus. Sketch Engine is software that helps to manage a corpus and analysis text, and It is developed by Lexical Computing Limited. Sketch Engine is a tool to collect text files of any language. It can access with the help of any browser. Millions and billions of words can be collected with the help of this tool. Based on the seed words, it will form a combination, and then it will search for the websites in the Bing search engine. Websites will be listed related to the seed words. Using the website, Sketch Engine will start collecting the words. Sketch Engine not only helps to get corpus, it also helps to give some additional information about the language, words, grammar, tokens, and many more. Steps involved in collecting the corpus using the Sketch Engine are: -

1. The “New corpus” tab is used to create the new corpus. Where you will select the language of the corpus, corpus type, and name.
2. In the next step, there are two options one is “Find text on the web”, which is used to find a text from the web ,and the other one is “I have my own texts”, which will allow us to upload our own words in a pdf file or txt file.
3. By clicking “Find text on the web”, we are able to give seed-words. Then using the seed-words, words will start collecting.

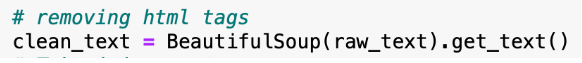
For my corpus, I selected Spanish in the language field. Clicked “Find text on the web”. In our group, we finalize some seed-words, and the seed-words are Sport – deport, player – jugadora (feminine), jugador (masculine), score – Puntaje, referee - Arbitra(feminine). Seed words are given in Spanish because this word will help to find the relative websites. My Top-level domain is “Co”. So all data will be collected from the relative domain. By using this, I have managed to collect 80,000 words in a corpus. Same as this, our groupmate also collected corpus for the different top-level domain (TDS). Each corpus has an average of 60,000 words. Before using this data in the classifier, we need to clean and understand the data in the corpus. Corpus will be in .txt file format. It will contain HTML tags, number, special character and blank space. Data present in it is unstructured. This is how the collected raw corpus looks like:



```
<p> Conozca a Pericles Cortez, árbitro del juego entre Oriente
Petrolero y Santa Fe </p>
<p> Cuando no arbitra se dedica a ejercer su profesión de odontólogo.
</p>
<p> Pericles Bassols Pegado Cortez es el árbitro brasileño que
dirigirá el partido entre Oriente Petrolero y Santafé. Tiene 41 años
y es internacional desde 2010, cuando no arbitra se dedica a ejercer
su profesión de odontólogo. </p>
<p> En su hoja de vida reposan apenas 12 partidos internacionales, 4
de ellos en Copa Libertadores todos los hizo el año pasado, el
promedio de tarjetas amarilla por partido fue de 6 y no expulso a
ningún jugador. En el principal torneo de clubes en Sudamérica no ha
dirigido nunca a equipos colombianos. </p>
<p> En Copa Sudamericana actuó como cuarto arbitro cuando Santa Fe
jugó como local en el partido de vuelta de la final de la
Sudamericana en la que se coronó campeón de la temporada 2015. </p>
<p> El único partido en su registro que aparece dirigiéndole a un
equipo colombiano en este torneo fue el 19 de agosto de 2014 y fue a
Águilas Doradas cuando jugaba en Pereira en esa oportunidad el cuadro
colombiano empató 1–1 con Emelec. </p>
<p> Es un arbitro que tiene buen estado fisco, algo muy difícil de
ver en los árbitros brasileños, se ubica bien en el terreno de juego,
posee un porcentaje alto en cuanto al acierto de la valoración de las
faltas, buena señalización y le da mucha continuidad a los partidos.
En la parte disciplinaria amonesta en el momento que es oportuno y
expulsa muy poco, razón por la cual tiene muchos problemas con el
manejo de los partidos. </p>
<p> En el papel hay garantías en la parte arbitral para los equipos
en contienda, claro, el señor Cortez tendrá que refrendarlo en el
partido. </p>
<p> A Cortez, en el estadio Ramón Aguilera Costas, de la ciudad de
Santa Cruz, le acompañarán por las bandas sus compatriotas Kleber
Lucio Gil y Rodrigo Correa, dos asistentes de muy buen recorrido y
experiencia en el ámbito. </p>
<p> Francia aplastó a Corea en el inicio del Mundial Femenino </p>
<p> Las anfitrionas se impusieron 4–0 en el partido inaugural del
certamen. </p>
```

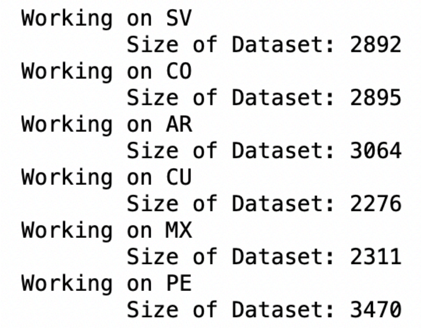
4 Data Preparation

After extracting the corpus, the next step was to delete unnecessary items from the corpus and reduce the corpus's size so that it could be faster and much more easy to classify. Raw corpus is unstructured because it has numbers, special character, HTML tags. By removing all these, we can convert these raw data into quality data. Since we are going to use these data to create a classifier corpus, it should be in a readable form. So we will create a text cleaning pipeline, by which text in the corpus is converted into tokens. HTML tags such as ‘<p>’ become the most frequent and visible item in the whole corpus, which may be an obstacle to classifying. NLTK (Natural Language Toolkit) is used to clean and prepare data in python language. As our corpus contain HTML tags that do not require for analysis part. First, we are removing it by using the Beautiful Soup library. Beautiful Soup is a python library that helps remove HTML tags and XML documents and extract data from the HTML. It integrates with your preferred parse to provide idiomatic navigation, scan, and modification of the parse tree.



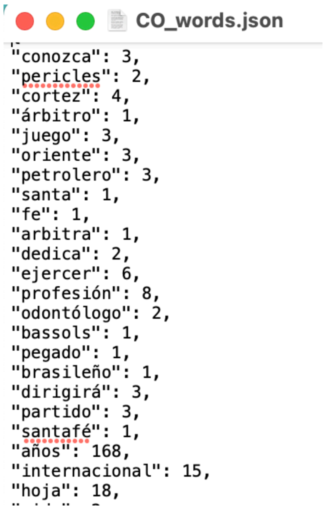
```
# removing html tags
clean_text = BeautifulSoup(raw_text).get_text()
```

We need to tokenize the text after deleting the HTML tags for further study. Tokenizing makes the corpus as a structured one by using the tokenizer library in python. Text in the corpus gets split separately. After that, by counting the unique words, we can find the size of the corpus.



```
Working on SV
        Size of Dataset: 2892
Working on CO
        Size of Dataset: 2895
Working on AR
        Size of Dataset: 3064
Working on CU
        Size of Dataset: 2276
Working on MX
        Size of Dataset: 2311
Working on PE
        Size of Dataset: 3470
```

In addition, the corpus contains several numerical values. These numerical values are not needed for the analyse part. So, we are removing all numerical values. Since the corpus was collected from many websites, it will contain many Stop words. These stop words will affect the accuracy and outcome of the classifier. Stop word is a set of common words used in many languages. To get better outcome, we are removing all the stop words. Duplicate terms are also removed. Now we had a formatted dataset with the following results in a data frame after deleting HTML tags, tokenizing sentences, and removing redundant words. The result will be in a JSON file with unique words with their word count.

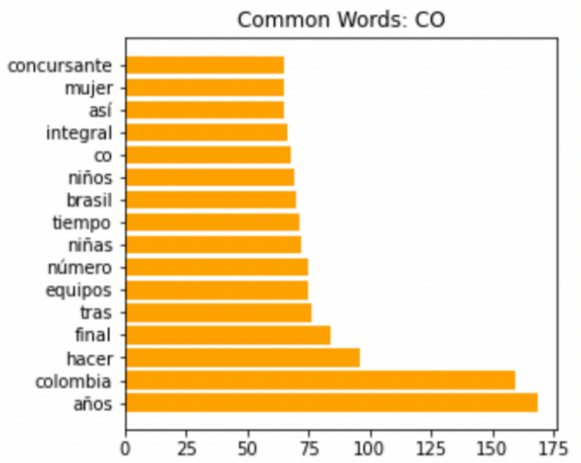


CO_words.json

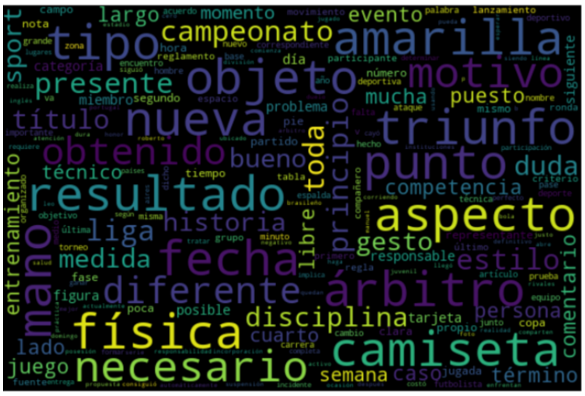
```
"conozca": 3,
"pericles": 2,
"cortez": 4,
"árbitro": 1,
"juego": 3,
"oriente": 3,
"petrolero": 3,
"santa": 1,
"fe": 1,
"arbitra": 1,
"dedica": 2,
"ejercer": 6,
"profesión": 8,
"odontólogo": 2,
"bassols": 1,
"pegado": 1,
"brasileño": 1,
"dirigirá": 3,
"partido": 3,
"santafé": 1,
"años": 168,
"internacional": 15,
"hoja": 18,
```

5 Data Modelling

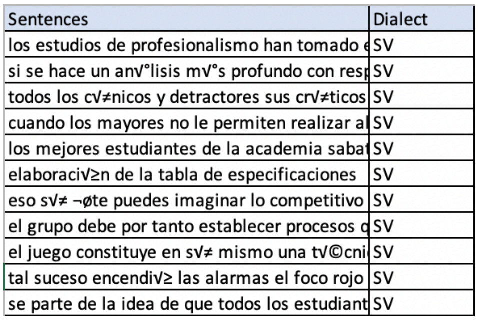
Pre-modeling is a critical step in the process of analyzing all of the process data. After completing all the process above, we will have unique words and their word count. Now we are mainly focusing on common words. We are going to train the classifier using these Common words. First, we need to sort the words according to the frequency to analyse with the common words. I attempted to map the top 15 most common words in my dialect against other dialects.



After plotting the frequency of common words in all Spanish dialects, I compared my 'CO' dialects with other Spanish dialects ('AR', 'SU', 'CU', 'MX', 'PE') and came to an understanding and overview of the frequency of common words across in the dialects.



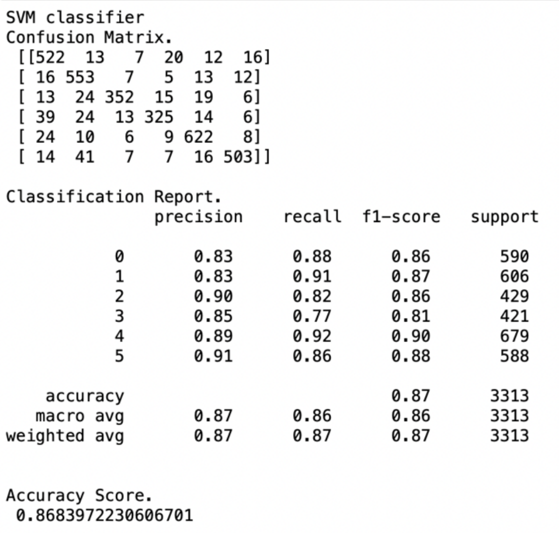
For the classification part, we are going to use text sentence for training. To create a readable sentence from the corpus, we are going to follow steps same as before. HTML tags are removed by beautiful soup, and URLs, punctuation, numbers from the sentence are removed. This is because the sentence may have some Gmail, formula or any other non-meaningful words. Those are not useful for the analyzing and classification part. Also, we are removing all the sentence which contains less than five words because most of the sentence below five words have less meaning. Duplicate sentences are removed. These things are done to increase the accuracy of the classifier model. All sentences are converted into lowercase and stored as a CSV file. As per the clean data CSV file has 16,563 rows.



Sentences	Dialect
los estudios de profesionalismo han tomado e	SV
si se hace un anv°lisis mv°s profundo con resp	SV
todos los cv≠nicos y detractores sus crv≠ticos	SV
cuando los mayores no le permiten realizar al	SV
los mejores estudiantes de la academia sabat	SV
elaboraciv≥n de la tabla de especificaciones	SV
eso sv≠ ¬øte puedes imaginar lo competitivo	SV
el grupo debe por tanto establecer procesos q	SV
el juego constituye en sv≠ mismo una tv©cni	SV
tal suceso encendiv≥ las alarmas el foco rojo	SV
se parte de la idea de que todos los estudiant	SV

In this experiment, I used the 'Sklearn' library, which imports a couple of machine learning classifiers such as Voting classifier, Random-Forest classifier, SVM classifier, MultinomialNB classifier, and GaussianNB classifier. WEKA is not used in this experiment because all works like data cleaning, data preparation are done in python language. So, we created classification part in python. Each dialect is labeled from 0 to 5 respectively.

For the SVM classifier, we are giving a confusion matrix before creating the classification report to achieve the desired accuracy. The result for the SVM classifier:



```
SVM classifier
Confusion Matrix.
 [[522  13   7  20  12  16]
 [ 16 553   7   5  13  12]
 [ 13  24 352  15  19   6]
 [ 39  24  13 325  14   6]
 [ 24  10   6   9 622   8]
 [ 14  41   7   7  16 503]]

Classification Report.
               precision    recall  f1-score   support

           0       0.83      0.88      0.86       590
           1       0.83      0.91      0.87       606
           2       0.90      0.82      0.86       429
           3       0.85      0.77      0.81       421
           4       0.89      0.92      0.90       679
           5       0.91      0.86      0.88       588

    accuracy                           0.87      3313
   macro avg       0.87      0.86      0.86      3313
weighted avg       0.87      0.87      0.87      3313


Accuracy Score.
 0.8683972230606701
```

With this method, dataset is trained for other four different classifiers. At the end of this experiment, we will find the optimizes classifier for the solution.

6 Evaluation

It's crucial to understand what each word in the classification table represents and how it's measured when analyzing the data. While classifying and training all five main classifiers method using a training set, we will get a classification report with the values of precision, recall, F1-score and accuracy. The voting classifier will use all the outcome of other classification models to improve the accuracy score.

Accuracy = (TP+TN) / (TP+TN+FP+FN)
Recall = TP/ (TP+FN)
Precision = TP/ (TP+FP)
Where: T - True, F - False

F1 Score = 2 * (Precision*Recall/ Precision + Recall)

After calculating precision, recall, and f1 score with the confusion matrix, we concluded that the Random forest classifier model has a minimum accuracy score of 0.56957, which is 57% accuracy. To improve the accuracy score, we used all model outcome in the voting classifier and got an accuracy score of 0.74434, which is 74% accuracy. This model has higher accuracy than other models.

Classifier	Accuracy
Gaussian NB	0.5901
Multinomial NB	0.704497
SVM	0.656505
Random Forest	0.569574
Voting Classifier	0.74434

In conclusion, different classifier models give different results. Classifier models are chooses based on the problem. Since the algorithm forecasts more on low model events. Accuracy is significantly better than both recall and precision when looking at the classification result.

Reference:

Alfaifi, A., Alfaifi, A., Atwell, E., & Atwell, E. (2016). Comparative evaluation of tools for Arabic corpora search and analysis. International Journal of Speech Technology, 19(2), 347–357. https://doi.org/10.1007/s10772-015-9285-5

Alshutayri, A., Atwell, E., Alosaimy, A., Dickins, J., Ingleby, M., & Watson, J. (2016). Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts.

Atwell, E., Arshad, J., Lai, C., Nim, L., Rezapour Ashregi, N., Wang, J., & Washtell, J. (2007). Which English dominates the world wide web, British or American? -- (Atwell, E et al., 2007)

Tarmom, T., Teahan, W., Atwell, E., & Alsalka, M. (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. https://doi.org/10.1017/S135132492000011X