Applied natural language processing

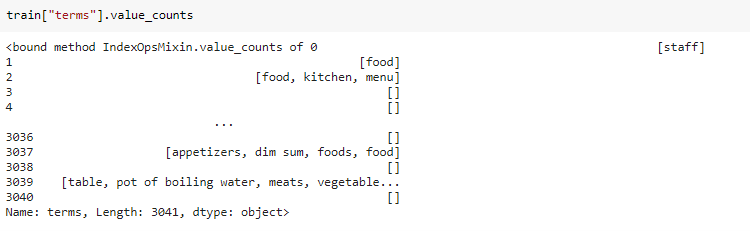
1. A description of the system you designed

The system was designed in the following manner. As it was known that the goal is to devise Dependency rules that will add descriptive terms, which will help in finding the polarity of the terms. So, it was believed that a bit of EDA(Exploratory Data Analysis ) is required.

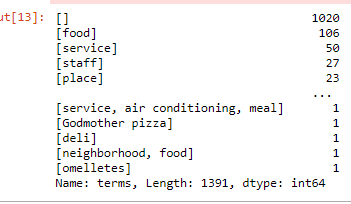
As required by the assignment, we need to extract the descriptive terms using the dependency parser ‘s pos(Part of Speech) tag.

EDA(Exploratory Data Analysis):

The screenshot below tells the Value\_counts of the terms.



Arranging these value counts in descending order this is what was got:-



Now, it can be seen that most of the text in the training data doesn’t have any terms- around 1020 of them to be precise. The interesting thing here is noting the part of the speech tag for the top 4 words – food, service, staff, and place.

Check-in in the dictionary helped to formulate a rule to get descriptive terms that will help to tell the polarity of that term. There were not many other factors to be considered.

* Pre-Processing :

Parsing XML File :



Was able to derive a function that could convert the XML file format into the desired data frame format that was needed. It was a bit exhausting to find a function or devise a function that gave us the data frame in the rightful manner that was required.

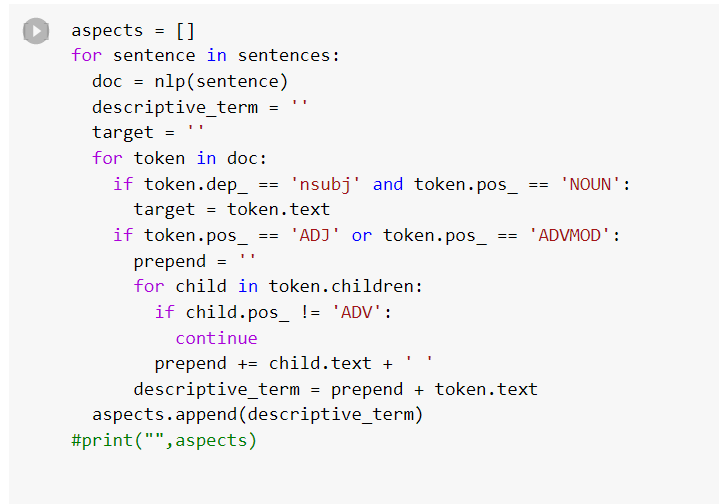
* Lemmatization :

I believe Lemmatization was not a helpful technique that was needed here. There are a couple of reasons why this is said. We are concerned with descriptive terms .i.e the terms that define the noun or a verb, we want to extract the actions. Therefore, shortening the word did not pay much heed to the process. For instance, the Naïve Bayes classifier that was built was used without lemmatization and with lemmatization, but the accuracy score was yet the same.

* Stop word removal: It was a conundrum to remove the stop words as it could be useful when building the rule. Where rule might check for conjunctions. Therefore, it was kept like that, not removed.
* Removing Punctuations and removing spaces – the function fix\_output() was used to remove these.

Lets us take into consideration the various rules that were considered during the project.

Rule 1: Finding the adjective Terms without the children being an adverb



The intention was to extract all possible descriptive terms, so it was important to

locate the nouns and the dependency tag for it was ‘nsubj’. Next, it was important to

extract all the adjectives. Though, a simple Rule –yields some results.

Rule 2: Verbs + Rule1 Terms + Amod Terms

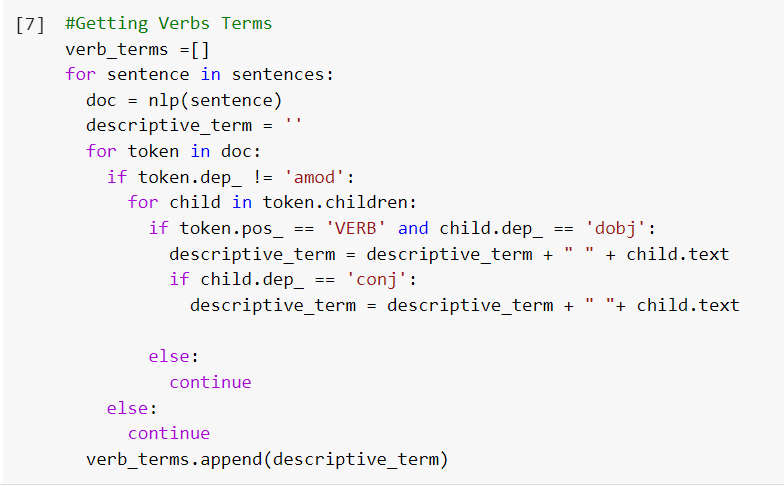


Figure 1: Verbs Extraction

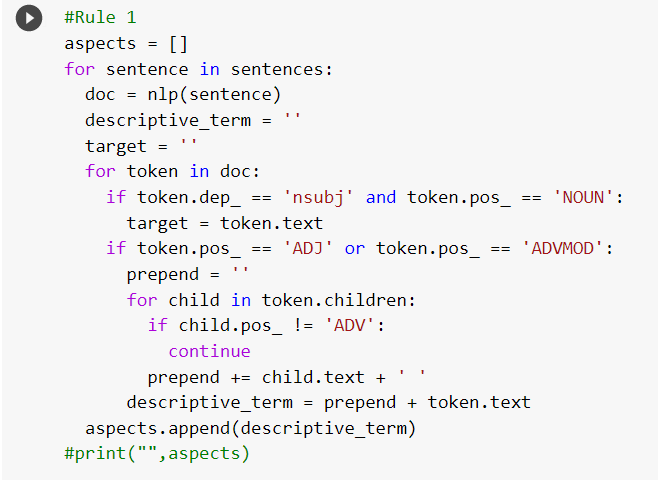


Figure 2: Rule 1 Terms



Figure 3: Amod Terms

These were the rules that were added.

A lot of trial and error was done, to find the tight rules for matching the desired descriptive terms.

Rule 1 was a naive approach but it did give way to understanding the best rules that can be matched.

It was needed that verb terms are important to extract, while Rule 1 gives the child token’s adjectives and the last stage of the rule matching was to get the adjective modifiers.

Other, parts of the rule were considered but once the desired out accuracy was received then, other rules did not help match the desired descriptive terms. However, let's have a look at the other sections of the rules:

* Open Causal Complement dependencies.



Figure 4: Open Causal complement.

The rule was formed carefully, if analyzed that the adjectives were considered but negation complement was not considered for the primary condition then auxiliary tokens were extracted.

* Adverb Modifier:



Figure 5: Adverb Modifier

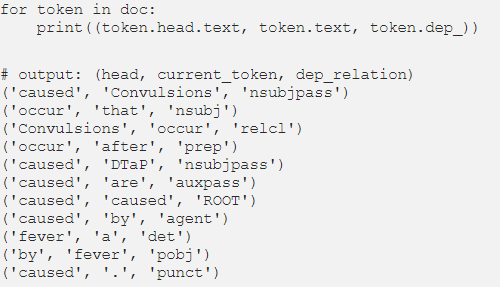
Adjective modifier was considered but it was considered as Rule 1, therefore it was found that the rule was redundant, so it can be left out.

* Shortest Depth Path:-

Upon research, I came across an article to find the shortest depth between two entities of the word. ( <https://towardsdatascience.com/how-to-find-shortest-dependency-path-with-spacy-and-stanfordnlp-539d45d28239>)

I did not implement this, rule as it was understood that want to take out the redundancies in a sentence, while I noticed that it extracts the adjective dependencies. Therefore, I used this logic to implement it as Rule 2 for this assignment. Through, my research it was understood that verbs and another part of speech tags might not be always important as the adjectives and the adjectives modifiers. So, I implemented it accordingly.

The Article shows the dependency of the path like below:



1. Analysis of various factors in your design. For example, you can compare the different sets of rules or different hyper-parameters/factors in your machine learning model. You are encouraged to use tables or figures to visualize the results

🡪

I was able to get the desired accuracy score.

The Factors that I tried:-

* Lemmatization and without lemmatization: For curiosity, I tried with and without lemmatization. The Accuracy did not change. It can be expanded and said the same works for stemming as well.

One would say, that what the reason for that it could be, the reason is that the word ‘running’ or ‘run’ is the same as the family of words. As the lemmatization uniformly does the distribution for all words. Another point to be considered is that this isn’t a real-time dataset where someone can write “run n ing”, as we are talking about a well-rounded dataset, so it makes sense that lemmatization did not impact the accuracies.

* Stop words removal :

Did not affect the accuracies, for a reason that the rule that was made did not extract conjunctions. Conjunctions like and, is, there, etc. The goal is to extract words that add some meaning to the aspect terms, so-called descriptive terms.

* Discussing the Rule 1:

I came across a site which explained the ABSA process in detail and they mentioned Rule 1 as a basic approach. However, there changes made as the article code was implemented to get the sentiment, while the goal of the assignment is to get the descriptive terms that when put into a classification, it classifies as positive, neutral, and negative. Therefore, it was understood that the emphasis should be given to the rule matching.

The Testing accuracies – can be seen did not improve as the number of testing accuracies that were there was 606, while the test dataset had 800 examples. Rule 1 did not consider other dependencies of the sentences.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rules | Model | Validation Accuracy | Testing Accuracy | Notes |
| Rule 1 | Multinomial NB | 0.51 | 0.75 |  |
|  | Logistic Regression | 0.55 | 0.59 |  |
|  | Random Forest | 0.52 | 0.58 | With Lemmatization/Without Lemmatization. No changes. |
|  | SGD Classifier | 0.41 | 0.48 | Linear Classifier with SGD training. |
|  | SVM | 0.53 | 0.59 |  |

* Rule 2 :

After many many trials and errors, many hours of research. The table below tells about the accuracies got from the models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rules | Model | Validation Accuracy | Testing Accuracy | Notes |
| Rule 2 | Multinomical NB | 0.53 | 0.87 |  |
|  | Logistic Regression | 0.50 | 0.91 |  |
|  | Random Forest | 0.55 | 0.90 | Y\_test 152., Predicted 152  606. |
|  | SVM | 0.45 | 0.91 |  |

The SVM and Logistic regression classifiers are models that categorize depending on the patterns, where our goal was to extract those words which these models can find the patterns for it, then extract them to the right polarity of the words.

* Removing conflicts:

Removing conflicts helped as the conflict sentiments were disturbing the accuracies of the model.

**References:**

<https://spacy.io/usage/linguistic-features#dependency-parse>

<https://intellica-ai.medium.com/aspect-based-sentiment-analysis-everything-you-wanted-to-know-1be41572e238>

<https://www.kaggle.com/manovirat/aspect-based-sentiment-analysis>

<https://towardsdatascience.com/aspect-based-sentiment-analysis-using-spacy-textblob-4c8de3e0d2b9>

<https://medium.com/analytics-vidhya/aspect-based-sentiment-analysis-a-practical-approach-8f51029bbc4a>

<https://www.kaggle.com/phiitm/aspect-based-sentiment-analysis/notebook>

<https://github.com/navreeetkaur/aspect-based-sentiment-analysis/blob/master/script/elmo.ipynb>

<https://towardsdatascience.com/how-to-find-shortest-dependency-path-with-spacy-and-stanfordnlp-539d45d28239>