Advance NLP

Assignment on Building an aspect-based sentiment analysis algorithm based on syntactic parsing

Manivannan Meenakshi sundaram

Uni Adeladie

a1842066

Due Date : June 17, 2022

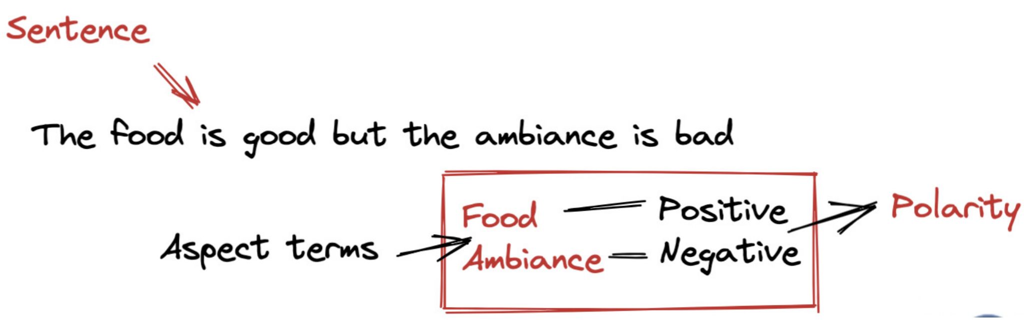
Assignment 3 - Syntactic parsing

The process of doing sentimental analysis in natural language processing is determined to automate their way of understanding the opinion or comment, whether it leans towards the positiveness or negativeness.

The analysis is done based on the study of the respective person's emotion, mood, attitude, and even their current state. Many active organizations use these types of methods to determine people's emotions and work on their product to give better solutions to the users. Companies gather feedback from users and automate it in a way to determine if that is a positive or negative sentiment.

ABSA :

In this assignment, we are asked to do sentimental analysis based on the aspect term in the sentence or a review given by a customer. Aspect-based sentimental analysis is one step further than traditional sentimental analysis by assigning polarity to the aspect terms which are possible in a review. For example, the ambience of the restaurant is good, but the food is horrible.



Understanding the given data :

The provided data consists of 3000+ restaurant reviews with their aspect category and aspect terms along with their polarity. The given format of the dataset is XML, which we need to convert it to a dataframe using python.

Review : This column has a wide range of reviews on restaurant based upon his/her emotions, attitude , current mood. People have written it in a different way where humans can see difficult to understand if that is a positive or negative.

Aspect Term : It is a phrase or a word which derives as a main attraction towards other words in a sentence

Aspect category : A category is a pre-defined collection of words where the aspect terms are categorized based on their approaches.

Polarity : Each aspect term is followed by a polarity. Its a sentiment which is derived from an aspect term based on humans emotions. It's mainly categorized in to three, namely Positive, Negative, and neutral terms. There is another category which is conflict which can be either, positive or negative or neutral, and we can neglect that in this assignment.

Aspect term extraction :

Inorder to recognize the aspect term which may be the key factor for the sentence which decides the polarity for a product, service, or food review. One of the common approaches to finding them is to extract nouns or noun phrases. These noun terms and phrases can act as a potential aspect term in most of the cases and also we can use some extraction rules to filter the terms which may not be an eligible aspect term. In our dataset the potential aspect term have been provided and we can make use of them in our analysis.

Aspect term polarity extraction :

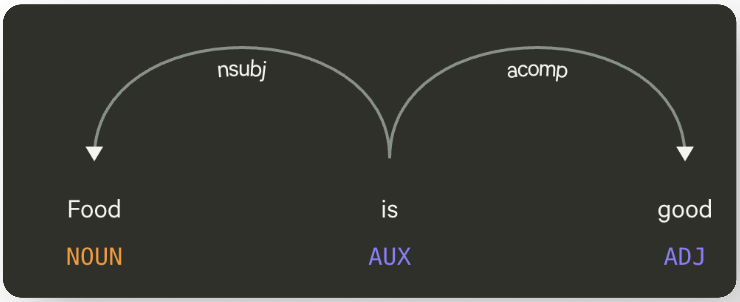
To find the dependency of the words from the dataset for the aspect term we are using spacy, a third party library for finding dependency of the words based on vocabulary, trained vectors, syntaxes, and entities.

Spacy :

Spacy allows the documents to get tokenized in to sentences and then further changes the tokenized sentences in to tokens, which can be accessed by iterating the documents

Parts-of-speech are the properties of the words which will be identified by the spacy and the most powerful feature spacy has is its dependency parsing from which we can iterate the child and grand child of the aspect term. The relations can be accessed by using the .root, .child, .ancestor etc.

For example, consider a sentence 'Food is good' the spacy parser for dependency is calculated as.



Approach for finding polarity :

There are different approaches for finding the polarity of the aspect term; some of the interesting methods which are used are explained below.

Rule based sentimental analysis :

Rule-based approaches are simple and created by humans based on the grammar rules available and that help to identify polarity. These set of rules combine many theories, aspects and calculates a score for the positive and negative terms for the words and defines the polarity of the sentiment.

Some of the rules used are :

- As given in the handout, if the token has an 'amod' then the polarity of the words gets a positive score, and it takes the polarity positive. Example : Having words like pretty, good, beautiful turns the words in to positive

- Token can also be an opinion word with the dependency 'advmod'. Example : He eats [amazingly] all the vegetables. Here amazingly acts as an Adverb and its an opinion word, so the sentiment moves towards positive.

- If the token has any 'neg' words, then the sentiment towards them completely pushes to the negative. Example : I [don't] like the food. Having negation terms like not, couldn't, etc will change the polarity to negative.

- If the token has any words related to negative word list, then the polarity changes to negative and has a score like ' 1.55' . Example : Abnormal, buggy, choppy, all are related to negative words.

- Then, Moving to the dependency children's of the token, and check for 'amod' or 'advmod'. If that's the case, sentiment towards the words gets affected.

- The above-mentioned rules have been applied for checking the child and grand children for the aspect term and the polarity has been calculated.

- If the score is neither positive nor negative, the polarity of the aspect term is considered as neutral.

Constrains and edge cases observed :

- Code could capture the polarity for most of the cases based on the aspect terms. But it was having difficulty to calculate the scores if there are many tokens which are related to the aspect term.

- The process of detecting the sarcastic reviews is way more complicated and the rules created were not able to detect the exact polarity.

- Model gets waived, if the token has some spell error or is not able to detect the dependency or aspect terms.

- The polarity of getting neutral is too low, which doesn't mean there are no neutral words. Aspect terms in the dataset have the polarity neutral.

- Rules for the polarity neutral are too general , that it will detect the polarity neutral if and only if there is no positive or negative terms.

- So, metrics only for positive and negative are calculated.

Performance Evaluation :

Based on the observations made, the built model could calculate scores for positive and negative at its best, and polarity neutral hasn't got any score.

The accuracy of the model gets increased based on the rules made. More the rules, More the accuracy score.

Here, I have calculated the scores of recall and precision for each sentiment based on the rules. I have neglected neutral sentiment as they have very low score.

These scores are calculated based on pairs of rules for each polarity.

Overall score for these rules came up to 67% for positive and 50% for negative.

```
Presicion Score For positive sentiment:
0.6109473684210527


Recall Score For positive sentiment:
0.6705175600739371


Presicion Score For negative sentiment:
0.509641873278237


Presicion Score For negative sentiment:
0.22981366459627328
```

Rules	Pos Recall	Pos Precision	Neg recall	Neg Precision
• Rule to detect amod and adverb	70%	57%	52%	59%
• Rule to detect neg and negative words	60%	67%	49%	20%
• Rule to detect child's amod	60%	69%	61%	20%
• Rule to detect grand child's amod	61%	67%	50%	22%

These are the scores obtained based on running particular rules on the model, and the scores were calculated based on that.

Success and failure cases :

In my case, the model could predict the sentiment of positive and negative words more precisely. For Example : Food is good and delicious The model returns a score of 1 which is positive. In other case, if the review is ‘Food is horrible ’ it gives a score of -1 which is negative.

If the aspect terms are words with siblings, the ability to find the polarity gets difficult and it returns null. Example : One of the reviews aspect term is ‘orrechiete with sausage and chicken’ and the polarity we got is {}

The model wasn’t able to predict neutral cases as more often it stacked with positive and negative cases.

Further implementations:

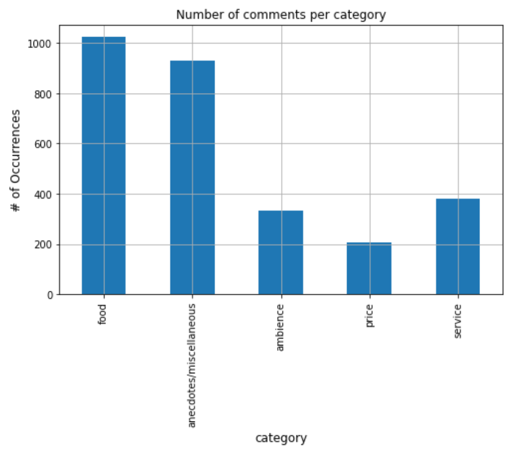
- I tried implementing TextBlob and sentimental analysis from vader. These thirdy party implementations have very much less score than the score which i got from the rules.

- The code can be viewed from this github link for this additional try-out

Aspect category extraction :

Aspect category is an additional extra problem which is often described as a text classification problem where the aspect terms based on the sentence are categorized into their respective category. Here we have 5 different categories, namely,

- Anecdotes/miscellaneous
- Food
- Price
- Service
- Ambiance



	sentence_id	text	anecdotes/miscellaneous	service	food	ambience	price
0	3121	But the staff was so horrible to us.	False	False	False	False	False
1	2777	To be completely fair, the only redeeming fact...	False	False	True	False	False
2	1634	The food is uniformly exceptional, with a very...	False	False	True	False	False
3	2534	Where Gabriela personaly greets you and recomm...	False	True	False	False	False
4	583	For those that go once and don't enjoy it, all...	True	False	False	False	False
5	2846	Not only was the food outstanding, but the lit...	False	True	True	False	False
6	1571	It is very overpriced and not very tasty.	False	False	False	False	False
7	1458	Our agreed favorite is the orrechiete with sau...	False	True	True	False	False
8	3161	The Bagels have an outstanding taste with a te...	False	False	True	False	False
9	2391	Nevertheless the food itself is pretty good.	False	False	True	False	False

In this we can see, below the fields there are bunch of terms, namely, true and false. True represents that it falls under that respective category, and false represents that it doesn't fall below that respective category. Based on this, we calculated scores for accuracy, recall, F1 and precision with respect to the test and train data for each category.

```
Processing scores for anecdotes/

 Accuracy is 76.39442231075697
 Precision is 89.8989898989899
 Recall is 28.164556962025316
 F1_score is 42.89156626506024
```

```
Processing scores for price

 Accuracy is 93.92430278884463
 Precision is 0.0
 Recall is 0.0
 F1_score is 0.0
```

```
Processing scores for ambience

 Accuracy is 88.74501992031873
 Precision is 100.0
 Recall is 0.8771929824561403
 F1_score is 1.7391304347826086
```

Appendix :

Full code can be viewed at this github repository