

# **EMOTION ANALYSIS OF REVIEWS**

## **OBJECTIVE**

In this project we have done sentiment analysis on Amazon Food Reviews. We have built an IR model wherein given an input review, our model will output a sentiment score (a value in [0,5]) for different aspects of the product.

## **DESCRIPTION**

The initial step is to decompose the given review (sentence) to micro-sentences (i.e. converting compound sentences to simple sentences). These micro-sentences are fed into the spelling correction module in which we have done Isolated Spelling Correction. The next step is to Tokenize the sentences of the entire dataset, after which we remove the stopwords from the list of tokenized words and the remaining words are then Lemmatized. As the list of words is too large, we have restricted our list to the most frequent 100 words. This completes our preprocessing step. Now, analysing these most frequent words and from general knowledge, we have formed equivalence classes of five categories/aspects : Taste Quality, Product Quality, Price, Quantity and Delivery. Under each of these classes, certain words have been mapped. For instance, 'healthy' and 'food' is mapped to Product Quality, while 'packaging' and 'delivered' were mapped to Delivery and similarly for others.

Now we map each micro-sentence of the given review to one among the five categories mentioned above using POS tagging, wherein important concepts of the micro-sentence were extracted and checked with the equivalence classes. For instance, 'food was great' is mapped to the Product Quality category. Similarly, this process is carried out for all the micro-sentences of the given review.

The final step is to extract the sentiment from each micro-sentence. Again, important concepts (adjectives/interjections) which weigh in more for scoring are extracted using POS tagging and scored using the SenticNet library. To make our model more robust, we have also maintained an emoticon dictionary (user generated) as well as a dictionary of words like 'very', 'wow' which add relative/absolute sentiment to the review. Both these dictionaries are used in increasing or decreasing the score of the micro-sentence. The score of each aspect is calculated by summing the individual micro-sentences that belong to its classes, and the final scores are normalised so that they lie between the range of 0-5.

An overall score and overall emotion of the reviewer is displayed which is an aggregate of the scores on individual aspects.

## **LIBRARIES**

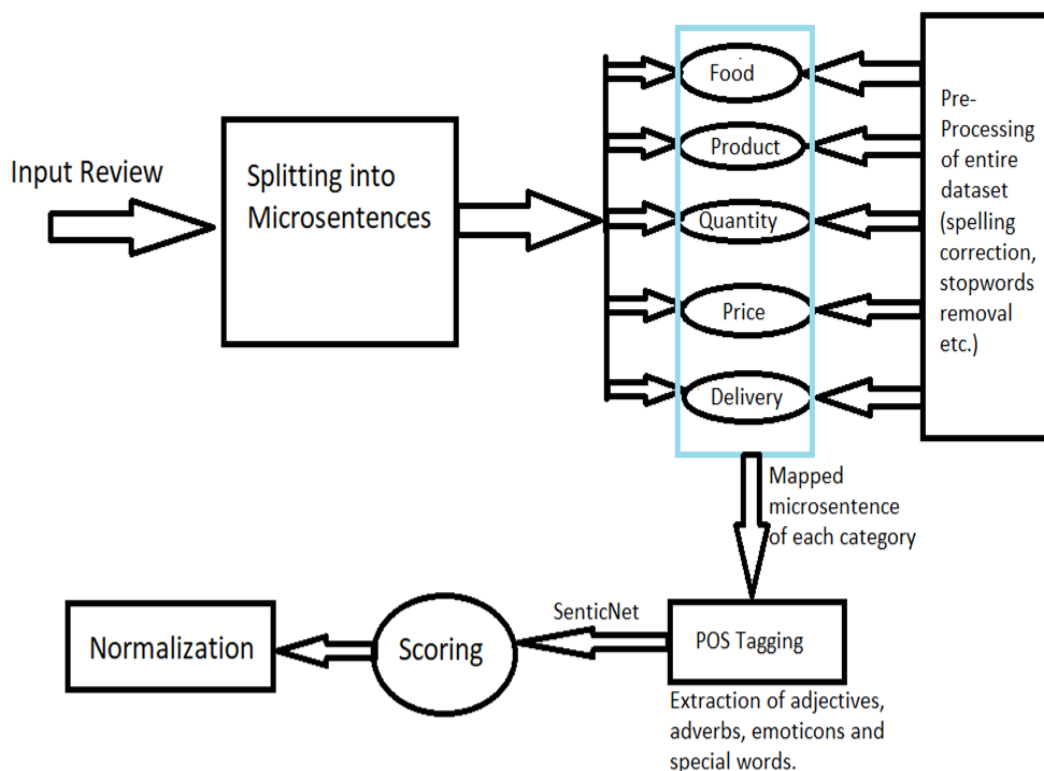
- Autocorrect - Python library used for Isolated Spelling Correction
- NLTK (Natural Language Tool Kit) - This library is used in the pre-processing step (Tokenization, Lemmatization, POS tagging, finding most frequent words)
- SenticNet - Python library used for sentiment scoring

## CONCLUSIONS AND EXTENSIONS

The model we have built successfully scores a review on input. We have used prominent Information Retrieval techniques in our model. But our model has not taken care of the following :

- Sarcasm - A sarcastic remark in the review can lead to false positives. For eg:- ‘What a great product!’ will be rated as a positive review, though the user might have sarcastically said.
- Context Identification - It is not fully accurate. For eg:- ‘The product was great and my dog loved it, but it could have been cheaper.’ Here, the second ‘it’ actually refers to ‘product’ but may get mapped to ‘dog’.
- Noise elimination - Random sentences irrelevant to scoring are present. For eg :- My dog loved this product. He has been eating this for the last 10 years since I brought my lovely baby home. Here, it may use ‘lovely’ to increase the score of the review.
- Inaccuracies in sentiment dictionary - The sentiment dictionaries available are not consistent with their scores of various sentiments. Wordnet performs poorly, while SenticNet although gives good results in most cases, still has a lot of inconsistencies. For eg :- SenticNet gives negative score to ‘very’.

## MODEL FLOW DIAGRAM



**Report by:**  
**Shivin Thukral (2015B4A70350G)**  
**Priyanshu Jain (2015B4A70379G)**  
**Aiswarya Subramanian (2015B4A70585G)**