**FEEDBACK ANALYSER**

* **Introduction:**

We look forward to build an application which will produce discernible visualizations to help end-users of this tool to collect comprehensive overview of product at glance on the basis of the feedback from the bunch of people who already hold the experience of using the particular product. We aim to provide the relevant visualizations of analysis to contribute in a considerable way in decision making.

* **Motivation**:

Our fundamental motive behind going for analysis of verified purchaser’s feedback is that the tool should be able to help out an amateur of the domain to buy the best product available in the spectrum. This application is solely built considering the mix-up mystification faced by the people to compare and analyze the product before purchasing. This perplexity is mainly because the feature description is available and even comparison between these descriptions is also available now-a-days, but one would always get more honest and practical reviews from the ones who already have used the product in day-to –day life.

* **Flowchart:**

Pre-processing of Data

Scrapping data from Web page

Classification of feedback

Feature Review

Overall product review

Ratings identified, overall emotion extraction, tone of the review analyzed

Sentence wise POS tagging, Sentiment analysis, keyword extraction

Aggregation of sentiments for features

Statistical computation

Data Visualization

* **Components explained:**

1. **Scrapping feedbacks from webpage:**

In this block the dataset is produced by collecting reviews of customer regarding a product and its features. The feedbacks are written in natural language. We need to use Web scrapper (or any other tool available) for scrapping data and dumping it in JSON format locally.

1. **Pre-processing of data:**

This block is required when the scrapper program fails to remove all kinds of tags or special characters in between the text. Here, the python script should be implemented accordingly to clean the dataset.

1. **Feature Review :**

* Sentiment analysis is performed for each sentence of each review for a particular product.
* The features and their sentiment score are stored in a dictionary.
* Special care is taken for aliases of feature name occurring in different sentences, so that all of them are pointed to one single category of feature.

1. **Overall Product review:**

* Rating out of five stars is considered for overall review.
* Along with the ratings, emotions like anger and joy are also given some weightage.
* This module also gives weightage to the tone of the title of each review.

1. **Statistical Computation:**

* The aim of this block is to provide average score for each product and feature wise scores for each feature of that product.
* This score produced will act as an input to next phase for plotting graphs.

1. **Data Visualizations:**

* Appropriate visualizations showing performance of the product using parameters like number of feedbacks, number of positive and negative reviews , overall score of the product, etc.
* **Technology:**

IBM Watson API’s

Python (version 2.7.9)

Watson Developer Cloud

* **API’s Used:**

1. **Natural Language Understanding:**

This tool is used to do sentiment analysis, POS tagging, emotion analysis and extraction of keywords from customer reviews.

**Usage:**

* The dataset in JSON format comprising of reviews on a certain product is used as input to the API
* The review text is then sent to Natural Language Understanding API
* The response is dumped as json object which is further used for analysis
* The tagging is done using the semantic\_roles component
* Emotion analysis is performed using the emotion component of NLU

**Challenges:**

* Sentiment of some negative sentences is given as positive

**E.g.** The phone heats up very much

This can be classified as a negative review foe the product but the overall sentiment is positive.

* Limitation of free plan on number of times the API is called
* Semantic roles component of this module provides subject, verb and object in response, but in some cases where multiple subjects or objects occur in sentences it fails to detect all of them thus hinder the mapping of features with their respective sentiments.
* POS tagging feature isn’t available and the POS tagger in Alchemy language is deprecated.

1. **Tone Analyzer:**

Used for tone analysis of the reviews through the title of customer review. This adds up to some part of the overall review score calculated.

**Usage:**

* Title is extracted from the dumped JSON object and sent to the tone analyzer
* The title tone score holds one fourth of weight in the overall score of the review

1. **Natural Language Classifier:**

This tool can be used to classify customer review sentences into features which are described in them.

E.g. If the sentence is "The camera quality is good" it gets classified into the feature camera.

**Challenges:**

But this requires a large dataset to train the classifier. We have currently not used it due to the lack of dataset available

* **Future Scope:**
* Natural Language Classifier can be used to train on sentiments.
* Accuracy in recognizing sarcastic comments can be used
* Static computations can be improved by using empirical data.
* An interface of chatterbot can be wrapped around.