# Application of Text Mining to Build a Recommendation System for Restaurants and their Dishes

Rafey Anjum
Pranveer Singh Institute of Technology
Kanpur U.P (208020), Dr.A.P.J Abdul
Kalam Technical University,
Lucknow U.P India

Harsh Dev, PhD
Pranveer Singh Institute of Technology
Kanpur U.P (208020), Dr.A.P.J Abdul Kalam
Technical University,
Lucknow U.P India

## ABSTRACT
Blogs, comments and reviews have now become an integral part of people who want to read them in order to be informed regarding other people opinion. This helps them to gain an overview of what other people say so that they might take a decision based on other people recommendation. Most of the time the user may not been in a position to read all the opinions and then take an informed decision about the product or services which he/she wants to take. Also it has been seen that most of the websites use different approach like star rating, numerical rating, to depict the information to the people who want to read the reviews. In this paper our aim is to develop a system for providing a method to help and explore good restaurants and specific dishes which a user wants to know based on past experiences of the people. The basic approach is to extract opinions from the websites and to extract the meaning of those sentences by applying Natural Language Processing techniques and then give the rating on a 5-point scale.

## General Terms
Text mining, Recommendation system

## Keywords
Natural language processing, Text mining, Recommendation system

## 1. INTRODUCTION
Many times the customers buy products and share their experience in their reviews[1,2]. The reviews are written in short para giving information about whole product and is features. Thus it conveys the sentiments about the product which they purchase [1]. This kind of text[1,4] documents have become popular to extract useful information. It becomes difficult for the customer to take the right decision in purchasing product with reference to data as there are thousands of reviews[1] about a product from the sources in the web.

This work like an automated tool to apply the text document from hotel review or blogs providing the best possibilities to find out good hold as well as good cuisines on four parameters i.e food, service, cost and

ambience. Though restaurants are given rating on a five point scale but users reviews have an essential role.

In general, the system will suggest a restaurant based on four parameters given by user and it will show the best dishes available at that restaurant.

In particular case if the customer is foodie with respect to cost So as per his review system will give weightage to food and less emphasize on cost. But one cannot guarantee correct reviews all the time while you are dealing with live data set .Currently recommendation system remain an active area of research. Some grammatical and punctuation blunders are bound to kept in so , we taking 700 to 1000 reviews per restaurant so that error rate could be minimized.

## 2. RELATED WORK
Opinion mining research started with identifying opinion defining words, e.g., great, amazing, wonderful, awesome, bad, poor, average etc. Later they shifted to star rating or thumbs up or thumbs down rating. But all these methods have failed to specify exclusive features on which a user expresses his satisfaction or dissatisfaction. Opinion summarization is the process of producing a sentiment summary, consists of sentences from reviews that capture the customers opinion on product features or objects on which customers have expressed the opinions. Opinion sentence identification has been carried out by many researchers by means of determining the presence of specific parts-of-speech such as adjectives, adverbs, etc. or a list of seed words that may potentially represent opinions.

Minqing Hu et al [1] work is considered as one of the pioneer work to find the summarization based on feature and opinion. They have used association rule mining to find frequent item sets, obtained from each sentence noun phrases. To prune the frequent items they have used different techniques. The infrequent features are identified based on the opinion word present in the sentence. Summary consisting of the product feature and the opinion about it has been given in terms of positive and negative.

Gamgarn Somprasertsri et al. [6 ] proposed an approach for mining product feature and opinion based on the consideration of syntactic and semantic information. They have used dependency relations and ontological knowledge with probabilistic model. They have also used Product Ontology to identify similar feature with different terminology.

The history of the phrase “sentiment analysis” is some respect equals that of opinion mining. Quite a few number of papers have been written on sentiment analysis and focuses on the specific application of classifying customer reviews on their polarity – positive or negative [6,7]. To obtain detailed aspects, feature-based opinion mining is proposed in literature [2,3,4,6]. Although, some opinion mining methods extract features and opinions from document corpora, most of them fail to capture the semantic relationship between the phrases used in the documents related to opinion mining.

## 3. PROPOSED APPROACH
The proposed approach has divided two main subpart. In first part is that we extract lot of characteristic from the given output. The second part is to cognize the view sentences specific to extracted output characteristic and find summary based users view.

The whole process is illustrated with the help of following

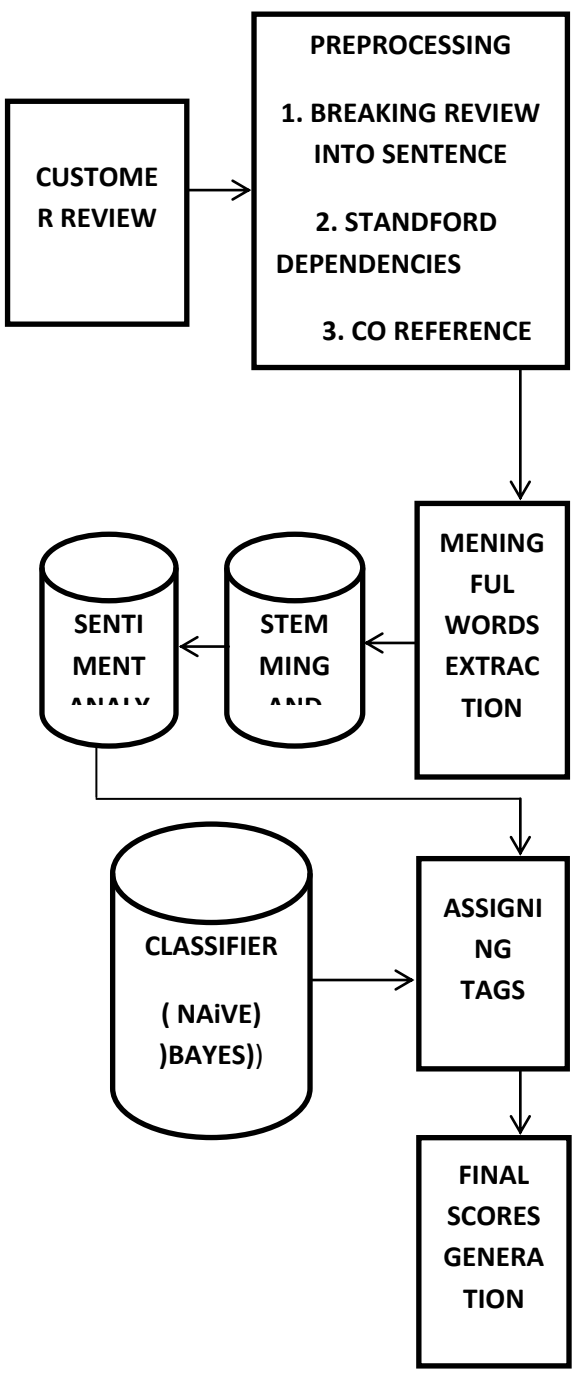


**Figure1**

### 3.1 Dataset preparation

The reviews for the project were taken from www.zomato.com

### 3.2 Extracting meaningful opinion pairs

After getting the reviews the main task was to get the meaningful words in order to do the analysis from a particular review.

Earlier we used python NLTK[1] to get the Named Entity Recognition technique to get the sentences and the sentiment [4]score of opinion words using SentiWordNet. But on finding that this system is not feasible Stanford parser was used to get the dependencies for sentences which were later parsed to get the meaningful words using the various rules.

The rules were based on the grammatical nature of sentences and the POS[1] technique. The dependencies were in binary relation between two words of a sentence and the relation could be noun-noun , noun-adjective , verb-adverb etc.

### 3.3 Generating scores

After getting meaningful results in a sentence , the sentiment dictionary for word was found by either using SentiWordNet or by using a lookup dictionary and scale the words from -5 to 5 (where -5 indicated most negative and 5 indicated most positive).

Now the classifier (naïve bayes classifier) was used to assign the words tags of food , ambience , service and cost. Hence after all the processing the review, we had a dictionary giving the extracted meaningful words and their scores.

after this the average was taken for all the reviews of a particular restaurant to assign them a rating for food, ambience, service and cost.

Preparing dataset by scraping reviews from food platforms like Zomato.

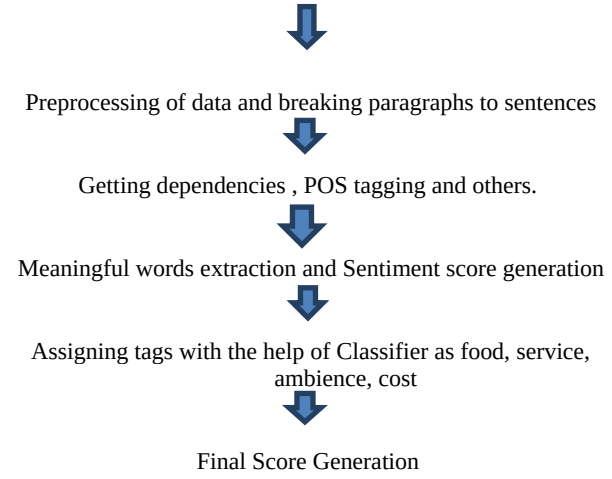


**Figure2:Flowchart depecting the flow of work**

### 3.4 Scrapping the reviews

A crawler was built using python requests module and beautiful soup to parse the html content to scrap the reviews and dishes dataset from the various websites including zomato.

The structure of the csv file looked like this:

```
Username,Review,Rating,Total_
Reviews,Followers,Expert level
    SundeepGupta,Rated wonderful
    experience nice place great ambience
    the terrace and lounge is amazing
    .nice food great environment Amazing
    place to spend time with the family
    and friends,Rated 5.0,1,0,??
```

### 3.5 Preprocessing

For example , if we had:

*We visited the TajMahal. It was very beautiful.*

Now it is a single sentence and the adjective beautiful referring to TajMahal could be resolved by Stanford parser.

This involved various tasks such as breaking paragraphs into sentences , co reference resolution etc. After this the sentences were fed to the Stanford parser to provide typed dependencies , POS tags etc.

```
Review :: The Butter Chicken was good but noodles were not good
------------------
Typed Dependencies
------------------
ROOT-->ROOT-0 good-5
det-->Chicken-3 The-1
compound-->Chicken-3 Butter-2
nsubj-->good-5 Chicken-3
cop-->good-5 was-4
cc-->good-10 but-6
nsubj-->good-10 noodles-7
cop-->good-10 were-8
neg-->good-10 not-9
ccomp-->good-5 good-10
-----------
POS Tags
-----------
{u'Butter': u'NN', u'good': u'JJ', u'but': u'CC',
u'were': u'VBD', u'not': u'RB', u'The': u'DT', u'noodles': u'NNS'
 u'was': u'VBD', u'Chicken': u'NN'}
```

**Figure3: showing typed dependencies and POS tags returned by Stanford parser**

## 3.6 Extracting candidate product feature

It is an adjective/s or a verb/s or a noun/s any of them. Lets see the dependencies which were used in this work to extract the opinion words

1. **nsubj** - Nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb. when the verb is a copular verb, the root of the clause is the complement of the copular verb[9]. Example: "The generator life is bad". nsubj (bad-6, generator-4) nn(life-4)

2. **dobj**- Direct object of a VP is the noun phrase which is the (accusative) object of the verb[9]."She gave me a raise" dobj(gave, raise)."They win the lottery" dobj(win, lottery)

3. **amod-** An adjectival modifier of an NP is any adjectival phrase that works to alter the sense of the NP.. [9]. Example; "Stephen makes orange drink."amod(orange-6, drink-5). The opinion pair is: (orange ,drink).

4. **advmod**- An adverb modifier of a word is adverb-headed phrase that works to alter the sense of the word[9]. " More seldom" advmod(seldom, more)

5. **conj**- In this two components connected by correlating cunjuction, such as "and", "or", etc[9]."Stephen is small and fake" conj(small, fake)

6. **dep-** A dependency is define as dep when the system is inadequate to apprize a more precise dependency relation between two words. [9]."Then, as if to show that he could, . . . "dep(show, if)

7. **neg**- It is the relation between a negation word and the word it modifies [9]."Stephen is not engineer" neg(engineer, not)

8. **root**- It represent the root of the sentences. The root node is indexed with "0", since the indexation of real words in the sentence starts at 1[9]. "I hate spring role ." root(ROOT, hate). "stephan is an loyal person" root(ROOT, person)

### *3.6.1 Rules to extract meaning*

Now let us look down the rules which were used to parse the dependencies using the above tags:

1. If the tag is advmod or amod or nmod then consider the pair as meaningful (their relative degree of meaningfulness will be decided later).

2. If the tags of the words in nsubj then consider them meaningful.

3. The tag compound is also taken into consideration to find out compound words. For example:

Butter chicken was delicious.

Compound(butter,chicken)

4. Find out the tags neg and conj for finding out the negative words appearing in the sentence.

Example: Ram is not a good boy.

Neg(good,not)

5. Root produces more ambiguous results and gives apt results only in some cases.

```
------------------
ROOT-->ROOT-0 good-5
det-->Chicken-3 The-1
compound-->Chicken-3 Butter-2
nsubj-->good-5 Chicken-3
cop-->good-5 was-4
cc-->good-10 but-6
nsubj-->good-10 noodles-7
cop-->good-10 were-8
neg-->good-10 not-9
ccomp-->good-5 good-10
------------------
Negative word list
------------------
[[u'good', 10]]
------------------
compound word list
------------------
{u'butter chicken': 1}

------------------
scores generation
------------------
good--->3.0
chicken--->0.0
good--->3.0
noodles--->0.0
```

**Figure4 showing parsing and results generated**

6. dobj tag is also taken into consideration , though with less relevence

Consider the sentence: The butter chicken was good but the noodles were not good.

### *3.6.2 Assigning weights to opinion words*

After finding out the meaningful words out of this, So we have used the following approach.

1. The noun – adjective (NN-JJ) pair was found to be most meaningful so is given the weight 1.

2. The noun-noun (NN-NN) pair was given the weight 0.8.

3. The noun-verb (NN-VB) pair was also given the score 0.8.

4. Other cases which included verb-verb or verb-adverb cases were given the weight 0.2.

### *3.6.3 Resolving Conflicts*

It is possible that the same word may be related to more than two words by some different relation. The following steps were taken for this:

1. How related the two words are to each other. The lesser the distance, the more they are related.
2. Otherwise weights assign to them as parameter to resolve conflicts. Hence the the pair with more weight is given more preference over the pair with less weight.
3. Hence with the above rules we were able to resolve conflicts to a large extent.

## 3.7 Getting Sentiment Scores

After getting the meaningful sentences the next task was to assign them scores which were found using NLTK's SentiWordNet or with the help of a lookup dictionary, containing more than 3000 words of noun, adjective , verbs etc. with scores ranging between -5 to 5.

The lookup dictionary performance was limited but faster than SentiWord Net. But before finding the scores the root words were found for the same.

**Stemming:** we use stemming with the help of WordNet Stemmer, through it we can change every noun an noun phrases which are in prular form, change into a singular form. For example:

Prettiest becomes pretty

**Negation identification**:

In this case the scores for food was 3.0 (good), but since it was identified as negative[6,7] the final score becomes -3.0.

Compound words: Therefore after getting the sentiment of a word if it is present as a part of compound word then it is replaced by the whole compound word.

- The TajMahal was beautiful.
- We get score as mahal -> 3.0 (for beautiful)
- But since tajmahal appeared as compound word , it is replaced as :
- Tajmahal -> 3.0
- After all this processing we get the following python dictionary :
- {word1:[score of word2,position of word1, position of word2,weight,word2]}
- Where word1 is generally a Noun and word2 is generally adjective which qualifies the noun.
- For eg . {food:[3.0,2,4,1,good]}

## 3.8 Assigning scores to target words

- For this task we used a Naive Bayes classifier trained on a dataset of more then 13000 short sentences or phrases.
- .The multinomial Naive Bayes classifier[7] is suitable for classification with discrete features (e.g., word counts for text classification. The accuracy was found to be more than 90% in assigning the tags. Hence the scores were changed as:
- {butter chicken:[3.0,2,4,1,good],waiters:[-4.0,3,6,1,annoying]}
- { butter chicken , waiters }

```
Assigning tags with classifier
------------------------------
{u'butter chicken': [3.0, '5', '3', 1, u'good', 'food'],
 u'noodles': [-3.0, '10', '7', 1, u'good', 'food']}
-------------
final scores
-------------

food-> 0.0
service-> -10000
ambience-> -10000
cost-> -10000
------------------
Recommended Dish
----------------
[u'butter chicken']
```

**Figure5 showing the assigning of tags by classif**

## 3.9 Results

After all these tasks, finally we are ready to generate scores for a review for the four attributes ie. Food, ambience, service and cost.

The concept of weighted mean was used to calculate the score.

$$x = \frac{\sum_{i=1}^{n} (x_i * w_i)}{\sum_{i=1}^{n} w_i}$$

The same procedure was repeated for other reviews of the restaurant. Since using weighted mean it is insured that the pairs with more meaning such as noun-adjective pairs get more part in deciding the overall average of a review. We could have also used the weightage to user who is writing reviews because zomato provides that in the form of users earlier reviews and if that person is a expert at that place. But to generalize the process this scheme was rejected because this information is not guaranteed to be present all the time.

If some tag is absent in a review then it was assigned a score of -10000 , so that it may not be considered afterwards. Finally the score was generated as the average of individual scores of the reviews.

### *3.9.1Recommending dishes:*

**Table1: Showing Final Score of Restaurant**

| s.no | User name | Food | Service | Ambience | cost | Recommend dish |
|---|---|---|---|---|---|---|
| 1 | 2,vaibhav agarwal | -0.18522727272 | -10000 | -10000 | -10000 | 4.5,[] |
| 2 | 3Shikha, | 4.1111111112 | -2.0 | 0.6666666666 | -10000 | 2.5.”[[u’dahi vada’,1]] |
| 3 | 4,Ashish sharma | -2.230769230769231 | 3.125 | 1.8571428571428574 | -10000 | 1.0,[] |
| 4 | 5,Toshi vasistha | 0.360000000004 | -1.0 | -0.7777777777 | -10000 | 1.0,”[[u’mysore masala dosa’,1]]” |
| 5 | 6,jasmeet singh | 1.33333333333333 | 2.46153846153846 | -10000 | -10000 | 3.5,[] |

For recommending a dish , found all the food tags with a positive score greater then or equal to 2 and weight greater then equal to 0.8 were taken.

After that the words were then searched in a dictionary consisting more than 4000 dishes. For near match , if a name consists of more then one word then all the words were searched separately in the dictionary and ascore was given on the basis of number of matches.

```
1 restaurant,food,ambience,service,cost
2 test_restaurant,1.67,1.87,1.94,1.7
3 cafe dalal street,1.57,1.903810058606032,1.3,1.08
4 jungle jumboore,0.23,1.8,0.5,1.6
```

**Figure 6 showing scores for different restaurants**

## 4. RESULTS GENERATION

After the scores dictionary is generated we are in a position to generate final scores with the help of weighted mean approach as discussed in the previous sections.

The csv contains the username and the score for ambience , service , food and cost . The value -10000 represents that the score for that attribute is absent. The dishes column contains the list of dishes being told by the user in that review in a positive sense.

**Table2: Showing Final Score of Restaurant**

| s.no | User name | Food | Service | Ambience | cost | Rating dishes |
|---|---|---|---|---|---|---|
| 1 | 2suhail Arora | 3.666666666 | -10000 | -10000 | 3 | 3.5[[u’rice’,1][u’mint mojito’,1][uçhilli potato’,1] |
| 2 | 3vaibhav agarwal | -0.1852272727 | -10000 | -10000 | -10000 | 4.5[] |
| 3 | 4shikha | 4.111111111 | -2 | -0.6666666667 | -10000 | 2.5[u’dahi vada’,1] |
| 4 | 5Ashish sharma | -1.2.2307692308 | 3.125 | 1.8571428571 | -10000 | 1[] |
| 5 | 6Toshi vasistha | 0.36 | -1 | -0.7777777778 | -10000 | 1[u’mysore masala dosa’,1] |

## 5. CONCLUSION AND FUTURE SCOPE

Text processing does not provide 100% accuracy in all circumstances. As the system is input text dependent so ambiguity prevails in some cases. Sentiword is not accurate in all situations for its high accuracy root word must be applied. words in review table must be accurately spelt. We are trying to increase the efficiency of classifier being used.

The present proposal gives an idea to the semantic analysis of any document without reference of whether the domain is question-answering, summarization or categorization, the extraction of noun phrases plays a important role. When use plain text as an input to standford parser it gives exact set of typed dependencies as output. The treatment of *det, nn, amod,* and *advmod* dependencies helps in the retrieval of compound nouns or noun phrases. Dependencies (*nsubj, dobj*etc) helps in extracting whether the noun phrases exist in subject or object role.

Process could be made faster with the help of multithreading.

New rules could be developed which could possibly extract the meaningful words from the dependencies more effectively.

## 6. REFERENCES

[1] Hu, M. and Liu, B.: Mining and Summarizing Customer Reviews, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), pp. 168 – 177, USA (2004)

[2] Liu, B., Hu, M., and Cheng, J. Opinion Observer - Analyzing and comparing opinions on the Web, in: Proceedings of the 14th International Conference on World Wide Web (WWW'05), pp. 342-351, Japan (2005)

[3] Kim, S. and Hovy, E.: Determining the Sentiment of Opinions, in: Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), pp. 1367-1373, Switzerland (2004)

[4] Popescu, A. M. and Etzioni, O.: Extracting Product Features and Opinions from Reviews, Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05), pp. 339 – 346, Canada (2005)

[5] 9. Pang, B. and Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, in: Proceedings of ACL'04, pp. 271-278, (2004)

[6] Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02), pp. 417 – 424, Philadelphia, Pennsylvania (2002)

[7] Pang, B., Lee, L. and Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02), pp. 79 – 86, USA (2002)

[8] Gamgarn Somprasertsri and Pattarachai Lalitrojwong "Mining Feature-Opinion in Online CustomerReviews for Opinion Summarization". Journal of Universal Computer Science, vol. 16, no. 6(2010),pp 938-955.

[9] Stanford typed dependencies manual.Marie-Catherine de Marneffe and Christopher D.Manning September 2008 Revised for Stanford Parser v.3.5.2 in April 2015