How to Cite:
Dandekar, P. R., & Pethe, Y. S. (2022). Theme extraction based on NLP. *International Journal of Health Sciences*, *6*(S5), 4885–4894. https://doi.org/10.53730/ijhs.v6nS5.9654

Theme extraction based on NLP

Prof. Pranali Rahul Dandekar
Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur
Email: dandekarpr@rknec.edu

Prof. Yoginee Surendra Pethe
Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur
Email: pethey@rknec.edu

***Abstract*---**Theme Extraction is a method of identifying, evaluating, and understanding human perception about a product in form of key features or themes that we extract dynamically from a given set of reviews within a data set. These themes are then categories to form an opinion about a given feature inside of the product through which we can analyze the advantages as well as the short comings of a given product or organization. These key features are then displayed on to the user for them to make a wise decision based on their likes and dislikes which they can compare with the user base that have already formed a review. All of this happens seamlessly with the help of Natural language Techniques that enables us to dynamically extract features or themes and generalizes an opinion score alongside it to represent thousands of reviews in a small concise manner. To make this happen, we consider seven different steps: (i)Text Pre-Processing, (ii) Removal of Stop-Words, (iii) Vader Sentiment Analysis, (iv) Feature Extraction using HAC, (v) Classification of Key Features using MOS, (vi) Testing the Accuracy of the Score and (vii) Creation of Word-Cloud using Features. After extracting all these features and scores from the review we can analyze and display them along side the reviews for better understanding of the product.

***Keywords*---**NLP, MOS, text processing, data set.

Introduction

This paper aims to make use of the highly unstructured text reviews about any product, extract the potential features about the product, assign scores to them and then classify the reviews as either positive or negative. The features of the product play a crucial role in the decision making process of the potential

customer. It is these features that distinguish one product from other similar products from different brands. Most of the things focus on a specific feature as their selling point.

Objective

The objective of this paper is when searched for a particular product on the web, the current day search engines show the list of websites which gives the features of the product and their prices. But the users can be given much more information about the product using the reviews about the searched product. Thus, a new type of theme extraction system can be designed which will not only retrieve facts but will also enable the retrieval of opinions of the users about the product.

Since the evolution of social networks, people have started to express their opinions in the form of the blogs or Facebook posts or tweets starting from the products people buy to the presidential candidate they support. When searched for a particular product on the web, the current day search engines show the list of websites which gives the features of the product and their prices.
But the users can be given much more info about the product using the reviews about the searched product.

Problem Definition

What if we could extract Key Themes from reviews dynamically without any reference?
And give Scores to the Themes based on the comments / reviews that are put by the users.
The project aims to make use of the highly unstructured text reviews about any product, extract the potential features about the product, assign scores to them and then classify the reviews as either positive or negative.

Proposed System

The Dataset we obtained contains unwanted text, and abbreviations.
Such a search engine can be used in several diverse applications like product reviews to aggregate opinions on a political candidate or issue. We create an NLP project which, when searched for a product, gives the highlighted features of the product and peoples opinion about the feature (instead of the user scrolling through all the reviews to know about the product).

Step-by-Step Procedure

4.3.1 Data Input:
We Collected Samples to Begin Testing.
4.3.1 Preprocessing:
Here Cleaning of the Dataset is performed.
4.3.3 Feature Extraction:
In this phase the actual required information is extracted.

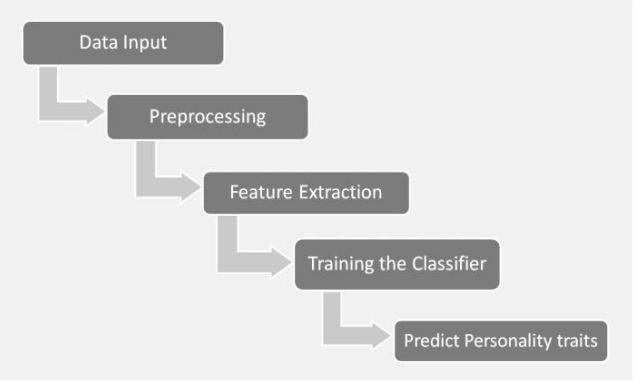
4.3.4 Rating Extraction:

Rate the emotion that is associated with each key themes and evaluate all reviews for the score.

4.3.5 Predict Traits:

The desired result is published.

Flowchart



Algorithm

Feature Extraction using HAC

1. The Algorithm HAC stands for High Adjective Count Algorithm.
2. To Extract Key Features, we implemented HAC Algorithm. Instead of using frequency of keywords, our algorithm starts identifying adjectives and nouns.
3. The scores of nouns are initialized to zero, each adjective is associated with a noun to which it is closest, this adjective is more likely to describe the noun.
4. For each such adjective, score of nouns is increased by one. After processing all the reviews, we will have a score associated with each noun, which we call them as opinion scores.
5. So, nouns with high score have more adjectives to describe them. Then we can have a threshold and nouns having score more than threshold are considered as potential features.

Single Linkage	$D_{12} = \min_{ij} d(\mathbf{X}_i, \mathbf{Y}_j)$	This is the distance between the closest members of the two clusters.
Complete Linkage	$D_{12} = \max_{ij} d(\mathbf{X}_i, \mathbf{Y}_j)$	This is the distance between the members that are farthest apart (most dissimilar)
Average Linkage	$D_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(\mathbf{X}i, \mathbf{Y}j)$	This method involves looking at the distances between all pairs and averages all of these distances. This is also called UPGMA - Unweighted Pair Group Mean Averaging.
Centroid Method	$D_{12} = d(\underline{x}, \underline{y})$	This involves finding the mean vector location for each of the clusters and taking the distance between these two centroids.
Ward's Method	$D_{12} = \sqrt{\frac{2.\lvert k \rvert.\lvert l \rvert}{\lvert k \rvert + \lvert l \rvert}} . \lVert \bar{x} - \bar{y} \rVert$	This method minimizes the total within-cluster variance. Those clusters are combined whose merger results in minimum information loss (ESS criterion).

Figure 5.1.4a: Formula associated with HAC.

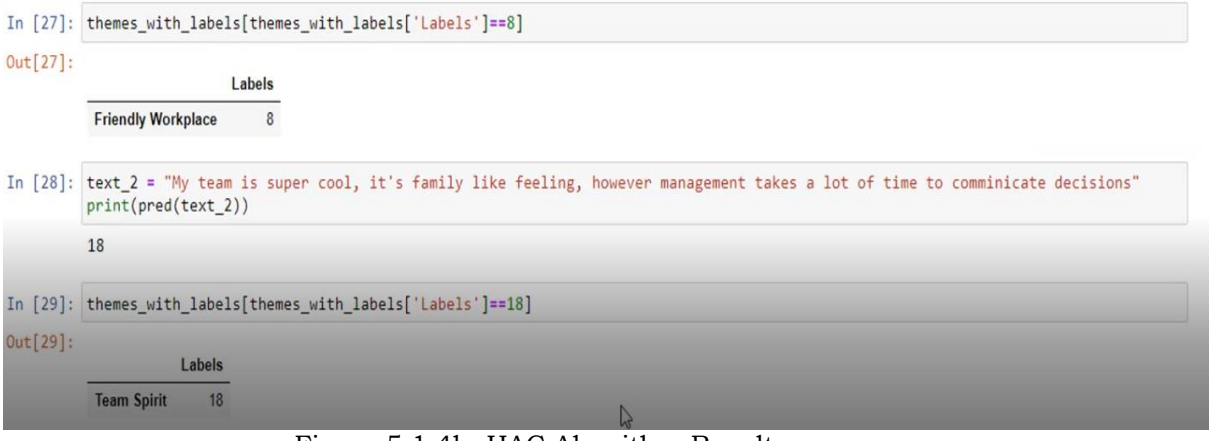


Figure 5.1.4b: HAC Algorithm Results

Classification of Key Features using MOS.

MOS stands for Max Opinion Score Algorithm.

1. MOS Algorithm Takes all the N-Grams and Bi-Grams identified Adjectives into account to perform Analysis on each sentence to look for words and analytically provide scores based on what is said.

2. This Algorithm takes 3 arguments as input:

– The first argument is the list of adjectives which are used to express opinions, we refer them as opinion words. We must choose a value manually between [-4,4] to each opinion word. A high score indicates a stronger opinion than lower score.

– The second argument is the list of inversion words like 'not' which give a negative sense to opinion, so when these words occur in the left context of opinion

words, they change the opinion sense. So, when an inversion word appears, we multiply the score by -1.
– The third argument is the list of potential features obtained by using TF / TF-IDF / HAC algorithm.3. The number of runs of non-zero columns will give the number of words or disconnected letters.
3. For each sentence, we look at the opinion words and identify features closest to it. The score of features is summation of scores of opinion words associated with it. The scores of features are further summed up to calculate score of review, so for each review we get a score and reviews are displayed based on this score.

$$Review\ score = \frac{\alpha . Title\ score + Body\ score}{\alpha + 1}$$

(α is the title weight co-efficient)

Figure 5.1.5: Formulae for MOS Algorithm

5.1.6 Testing the Accuracy of the Scores
Tested the program on 5 different datasets.
The Average Accuracy came to **93.8%**.
The Dataset Ranges from 700 reviews to 25000 reviews.
Dataset were from different categories including.

- Jungle Resort
- Bank Employee Audit
- Digital Camera
- Television
- Internal Employee Reviews

	A	B	C	D
1		**Theme**	**Score**	**Percent**
2	**1**	view	3.2	80
3	**2**	exposure	3	75
4	**3**	Place	2.857	71.425
5	**4**	coordinati	2.8	70
6	**5**	order	2.8	70
7	**6**	approval	2.8	70
8	**7**	experience	2.8	70
9	**8**	SUPPORT	2.8	70
10	**9**	insurance	2.8	70
11	**10**	balance	2.8	70
12	**11**	rewasie	2.8	70
13	**12**	care	2.8	70
14	**13**	pay	2.8	70
15	**14**	Training	2.8	70
16	**15**	place	2.643	66.075
17	**16**	workplace	2.533	63.325
18	**17**	quality	2.5	62.5
19	**18**	communic	2.5	62.5
20	**19**	policy	2.4	60

Figure 5.1.6: Example of the Final Scores

Creation of Word-Cloud using Features

A Word-Cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color.

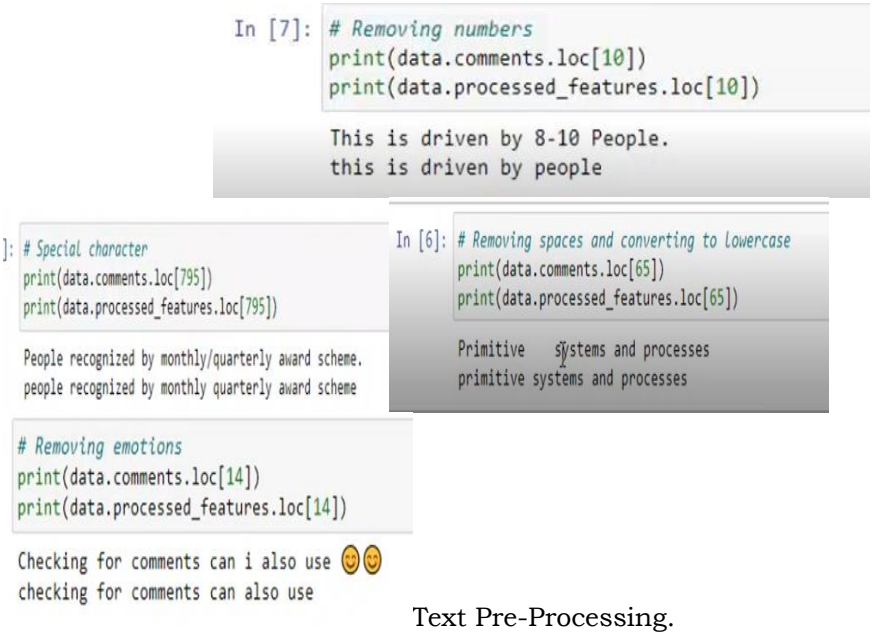
Figure 5.1.7: Word-Cloud Creation using Features.

Implementation

Text Pre-Processing

To extract key Features from the dataset, the following operations are performed. All these tasks are performed with module Re.

1. Removal of Numeric values. Unwanted numeric values do not provide any significant benefit as the program cannot understand nor evaluate it.
2. Removal of Special Characters (/, @, etc.). Unwanted special characters do not provide any significant benefit as the program cannot understand nor evaluate it.
3. Removal of Multiple Spaces. Unwanted spaces do not provide any significant benefit as the program cannot understand nor evaluate it.
4. Converting all the values in the text to be in lowercase as it is not only easier but all significantly beneficial for the program to understand each word by it own. Higher case letters inside a word are generally misunderstood and provide inaccurate results.



Text Pre-Processing.

Removal of Stop-Words

Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc. Such words are already captured this in corpus named corpus. We first download it to our python environment.

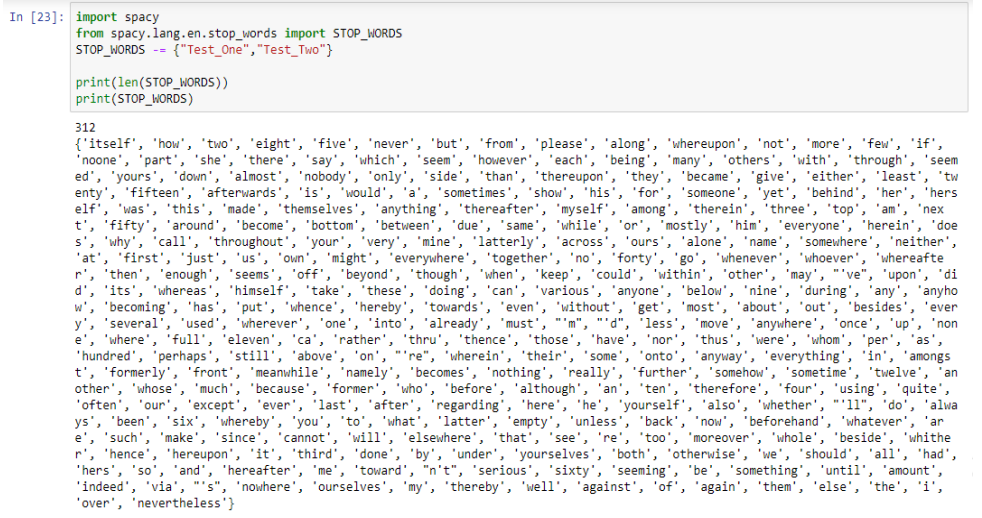
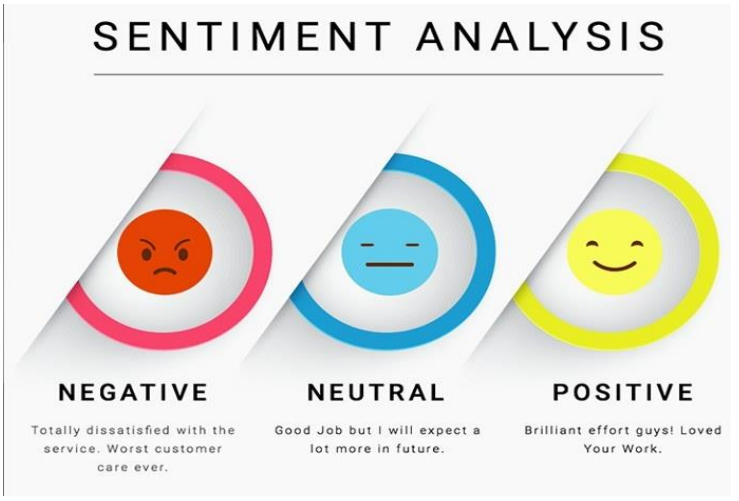


Image with Stop-Words.

Vader Sentiment Analysis on Reviews

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. ... VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as senti ment scores.



Vader Analysis Example.

Conclusion

With the proliferation of social networking and e-commerce the information contained in the opinions/reviews expressed by the people has grown by leaps and bounds. In this work we present an opinion search engine system that incorporates two novel opinion mining algorithms. The opinions are based on features and the orientation of these opinions is also largely based on the features rather than a product. People seem to like/dislike a specific product because of some feature associated with the product. The proposed framework not only classifies a review as positive or negative, but also extracts the most representative features of each reviewed item and assigns opinion scores on them. An initial experimental evaluation on several customer review data sets has shown that our algorithm achieves very high levels of accuracy.

7.2 Future Scope

Our plans for future work include experimenting with datasets from other social media. We also plan to further explore the idea of focusing on particular parts in a user's expressed opinion (e.g., the parts that are being most commented on) and extract features from there instead of the whole text.

References

[1] D. Winder, The importance of social mobility, Gemalto Review Magazine (2010).
[2] V. Hatzivassiloglou, K.R. McKeown, Predicting the semantic orientation of adjectives, in: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL'98, 1997, pp. 174–181.
[3] P.D. Turney, M.L. Littman, Measuring praise and criticism: Inference of semantic orientation from association, ACM Trans. Inf. Syst. 21 (2003) 315–346.
[4] H. Kanayama, T. Nasukawa, Fully automatic lexicon expansion for domain-oriented sentiment analysis, in: Proceedings of the 2006 Conference onEmpirical Methods in Natural Language Processing, EMNLP'06, 2006, pp. 355–363.
[5] Rahma, D. Y., & Atmaja, M. H. S. (2022). Peritoneal carcinomatosis and mimicking on CT scan findings. International Journal of Health & Medical Sciences, 5(1), 154-162. https://doi.org/10.21744/ijhms.v5n1.1864
[6] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2 (2008) 1–135.
[7] Rinartha, K., & Suryasa, W. (2017). Comparative study for better result on query suggestion of article searching with MySQL pattern matching and Jaccard similarity. In 2017 5th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE.
[8] A. Esuli, F. Sebastiani, Determining the semantic orientation of terms through gloss classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM'05, 2005, pp. 617–624.
[9] S.-M. Kim, E. Hovy, Determining the sentiment of opinions, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING'04, 2004.
[10] A. Meena, T.V. Prabhakar, Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis, in: Proceedings of the 29th European Conference on IR Research, ECIR'07, 2007.