Sentiment Analysis and Extractive Summarization of Amazon Review Dataset

Dr. Rajendra Roul
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Aditya Agarwal
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Akash Gupta
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Atul Shanbhag
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Kathan Kashiparekh
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Saurabh Shekhar
Dept. of Computer Science
BITS Pilani, K.K.Birla Goa Campus

Abstract—**With the increase in presence of online shopping companies like Amazon, E-Bay and Flipkart, people give a wide range of reviews for the products they purchase. Some are too long, some too short, some difficult to understand while some are totally irrelevant. Thus, there is a pressing need to reduce the diversity of reviews for a particular product and show the users only the most useful and most important of reviews about a product. In this work, we aim to summarize the reviews for movies purchased from Amazon using a combination of already developed algorithms and a feature selection technique. Sentiment analysis has been performed to categorize reviews into positive and negative. We have used a novel method for extractive summarization known as hierarchical summarization to summarize large reviews into summaries having few number of sentences. The results of this summary are compared to the existing algorithms using the ROUGE score to determine the best summary.**

I. Introduction

With an ever increasing trend of online shopping, people tend to rate their purchases by giving reviews of a product on the e-commerce companies websites. These reviews are seldom informative along and are quite large in number. Thus there is a need to combine these reviews and represent them in a short summary by extracting the important essence of the reviews. This technique of summarizing documents into short summaries is known as Text Summarization. Traditionally, Text Summarization algorithms are classified into two categories. Extractive Summarization is where the sentences in a document are ranked based on their relative importance in the document and the top such sentences chosen. Abstractive summarization on the other hand summarizes the docoument in its own words analogus to how we summarize some event to our friend in our own words. In order to achieve this, Abstractive summarization requires lots of data and complex algorithms thereby making it a difficult task to achieve.

However, in order to judge the accuracy of the generated summaries, we have human- generated summaries which are summaries generated by humans according to what they found important in a document. The predicted summary is then compared to these human generated summaries to check for how much important information is being retained by the predicted summaries. But in the real world, it is difficult to find human generated summaries for every dataset that we find because it is a tedious and cumbersome procedure to generate these summaries. Especially in the case of Amazon reviews, since the reviews are written in an unstructured manned it is quite difficult to generate human summaries for such a large number of reviews.

Thus, in order to overcome this problem we have implemented a novel approach to summarizing Amazon movies review dataset using a technique that we call Hierarchical Summarization which is an extractive summarization technique, which is described in further sections. Four different extractive summarization algorithms have been used to generate summaries from which important sentences are extracted using a feature selection technique to generate the human generated gold summaries. Prior to all this, we classify data into positive and negative reviews using sentiment analysis to improve the summarization of reviews and make it easier for the user to get detailed information about a product. The work in the report is as follows. In Section 2 we talk about the experimental design where in we describe each step of the pipeline used to generated the gold summaries using hierarchical summarization and the reason for using a particular technique. In Section 3 we present our results and an analysis of the summaries by comparing it to benchmark algorithms using the ROUGE score. In Section 4 we conclude our work and propose direction for future work.

II. Methodology

A. Data Collection

This work has been done on the amazon review dataset [1]. The dataset contains reviews of movies and TV shows purchased from amazon. The attributes of the dataset are reviewerID, asin, reviewerName, helpful, reviewText, overall, summary, unixReviewTime, reviewTime. Some of the important attributes include reviewerID : unique ID of the reviewer, asin : unique ID of the movie, reviewText : The review given by the reviewer corresponding to the reviewerID for the product with unique ID as asin, summary : The heading of the reviewText.

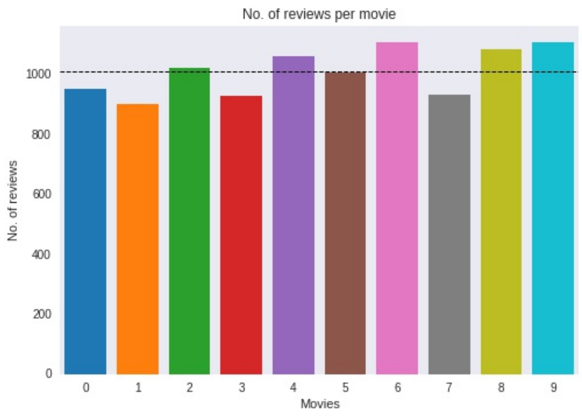


Fig. 1. Number of Reviews for the top ten selected movies.

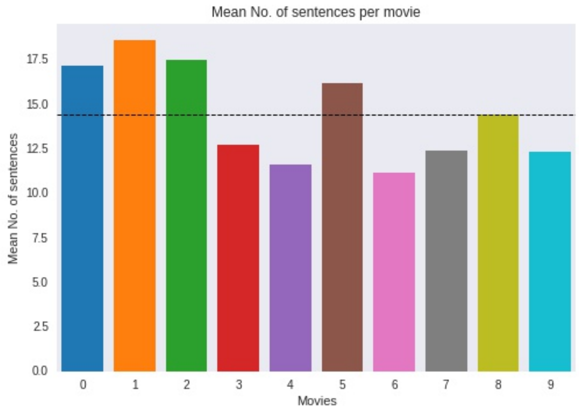


Fig. 2. Mean number of sentences per review.

B. Data Preprocessing

In the Amazon Review Dataset, since the reviews are user generated, the data is highly unstructured. There was no general structure to punctuations, spaces etc. As a result extensive preprocessing of the data was done before further use for obtaining accurate results. We used various regular expressions and tokenizers to convert the data into a well-defined format. Keeping in mind, our aim to perform sentiment analysis and extractive summarization we selected the top ten movies based on the maximum number of reviews. On an average each selected movie had thousand reviews with average review length being two hundred fifty words each which comes to around 20 sentences each.

The plot in Fig. 1 shows the average number of reviews for the selected movie set stands at around 1000. These were the movies with highest number of reviews in the Amazon dataset that we chose for analysis.

The graph plotted in Fig. 2 and Fig. 3 elaborates the fact that the selected movie-set had an overall 15 sentences per review per movie along with 250 words per review on an average.

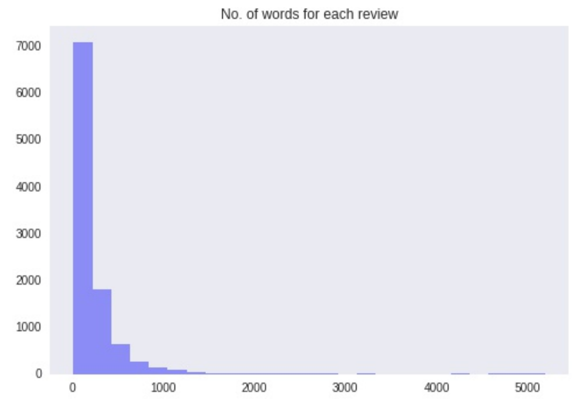


Fig. 3. Distribution of number of words in reviews

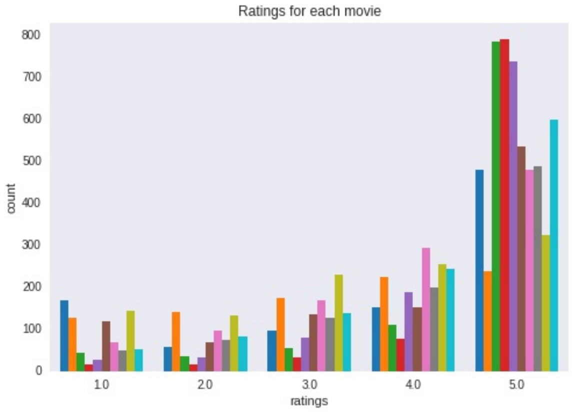


Fig. 4. Distribution of ratings of movies given on a scale of 1 to 5.

The movies have been selected in such a way that there is enough data available, comparatively, for analysis to generate out the positive and negative classes for the movies, as shown in Fig 4.

C. Sentiment Analysis

The reviews have been categorized on the basis of the polarity into positive and negative classes by carrying out a sentiment analysis on the pre-processed dataset. To achieve this a pre-trained Naive Bayes classifier on a movie reviews data was used which was provided by the TextBlob library. The model was trained on features like number of positive and negative words which help in determining the polarity of an input review/sentence.

$$P\left(A|B\right) = \frac{P\left(B|A\right)P\left(A\right)}{P\left(B\right)}$$

Here, A is the the class (positive/negative) that the review B will be classified into on the basis of the features that B contains and have been learned by the model.

```
Data: Pre-processed reviews for various movies
Result: Two lists of positive and negative pool of reviews
positive_list ← [] ;
negative_list ← [] ;
for each review in PreProcessedReviews do
    polarity ← sentiment_analysis(review);
    if polarity > 0 then
        add review to positive_list ;
    else
        add review to negative_list ;
    end
end
return positive_list, negative_list ;
```

Algorithm 1: Preprocessing

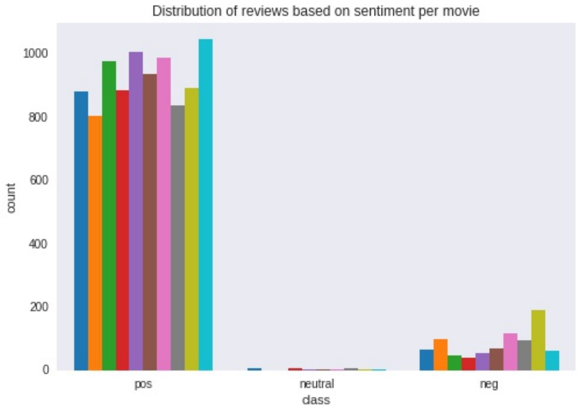


Fig. 5. Sentiment Distribution per movie.

After doing the sentiment analysis on the pre-processed dataset, the distributions of reviews are as shown in Fig. 5. We discarded the neutral reviews, as our main focus was to concentrate on the top positive and negative aspects of the movie, and do the analysis on the same.

D. Hierarchical Clustering

The following four algorithms have been used for hierarchial clustering -

- $LexRank$[1] : LexRank is used for computing sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. In this model, a connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences.
- $TextRank$[2] : This algorithm is a graphical text ranking algorithm in which text units best defining the tasks are added as vertices in the graph. The relation between the text units is represented as edges in the graph. After this graph based ranking algorithm is run until convergence and vertices are sorted based on their final score.
- LSA[3] : Latent Semantic Analysis is an algorithm of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. It helps in reducing the dimensions by selecting the most relevant features using SVD (Singular Value Decomposition).
- $SumBasic$[4] : This algorithm uses the probabilistic approach in which sentence weight is calculated by taking the mean of the probabilistic distribution of the words appearing in that sentence. Best scoring sentence containing the highest probability word is selected.

Pseudo-Code

```
Data: List of reviews
Result: Summarization of list of reviews in 20 sentences
initialization;
final_extractive_summary ← [ ] ;
one_sentence_summary ← [ ] ;
for each review in ListOfReview do
    temp ← review;
    while numberOfSentences(temp) > 1 do
        length ← numberOfSentences(temp) ;
        temp ← SummarizationAlgorithm(temp, length/2)
          ;
    end
    add temp to one_sentence_summary;
end
filtered_summary ← [ ] ;
for each review in one_sentence_summary do
    polarity ← sentiment_analysis(review) ;
    if |polarity| > threshold then
        append review to filtered_summary ;
    end
end
while numberOfSentences(filtered_summary) > 20 do
    length ← numberOfSentences(filtered_summary) ;
    filtered_summary ←
      SummarizationAlgorithm(filtered_summary,length/2)
      ;
end
return filtered_summary ;
```

Algorithm 2: Hierarchical Procedure

Above is our novel summarization procedure, Hierarchical Summarization that we run using each of the four above mentioned algorithms for the positive and negative collection of reviews to get four summaries of 20 sentences each which are the system generated summaries that we use for comparitive analysis later.

The threshold for polarity was set to 0.5 for getting featured in one of the positive or negative lists. This was done to ensure that the summarization is biased to the most positive and negative reviews.

The main motive behind using Hierarchical Summarization was to ensure the prevention of data loss that would have occured had we used each algorithm for summarization in one go. Also, it was found that summaries generated using this technique which were rich in information and thus gave dependable results for our specific dataset.

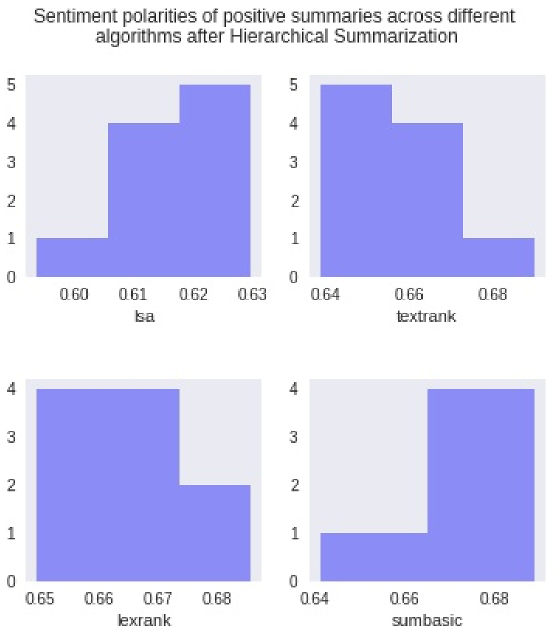


Fig. 6. Segment Polarities of Positive Summaries across different algorithms after Hierarchical Summarization

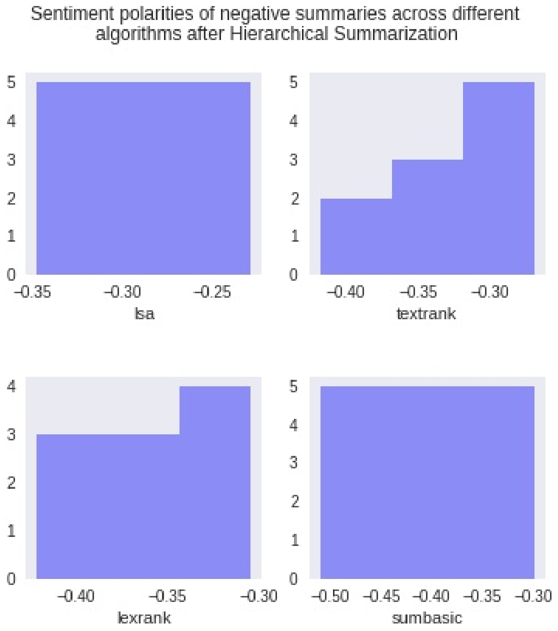


Fig. 7. Segment Polarities of Negative Summaries across different algorithms after Hierarchical Summarization

It can be seen from Fig 8 that applying summarization just once gives an average sentiment score of 0.2. However, as the study is more focused upon getting the best positive and negative summaries, it is necessary to choose the best sentences at each step. Thus after applying the Hierarchical Summarization, we can see that average sentiment score for positive reviews is around 0.66 (Fig 6) while for negative reviews it is around -0.35 (Fig 7). This captures the essence of motive behind the study.

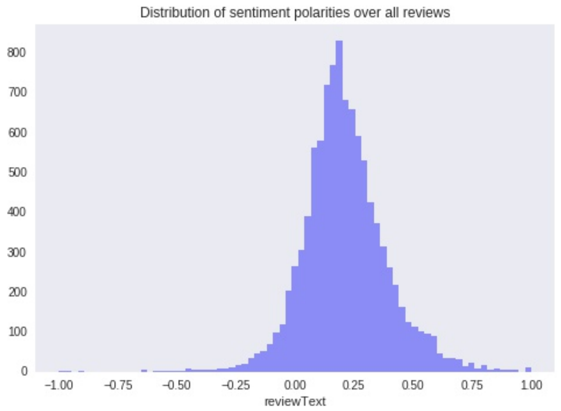


Fig. 8. Distribution of Sentiment Polarities over all reviews

E. *Feature Selection*

In order to compare the results more effectively, a benchmark was necessary which could mimic the human written summary. To achieve this, the four summaries obtained above were concatenated and a novel feature selection technique applied to obtain the 20 most important sentences for each of the positive and negative class. Some of the important features used to rank the sentences for selecting the top 20 sentences are as follows -

1. Length of Sentence First the normalized length of the sentence is calculated. The sentence is awarded less score if its length is very small or very large. The score has been calculated by modeling the distribution on the parabolic curve.

2. Positional Feature If the sentence is coming at the start of the sentence or it is coming toward the end then it has been given more score. This has been done keeping in mind the increased amount of importance given to the introduction or conclusion in any piece of text.

3. Weight of Sentence First the TF-IDF (Term Frequency-Inverse Document Frequency) weight of all the words was calculated, and depending on the frequency of the word in the sentence, the weight of the sentence is calculated.

4. Quoted Text If the quote exists in the sentence then the score of one is given, otherwise a score of zero is given.

5. Upper Case Text Sentences containing more uppercase words have been given more weight. This has been done as the upper case words represent the sentiments of the users more strongly.

6. Sentiment Polarities The absolute value of the sentiment polarity of the sentence on the scale of zero and one is taken as the score of the sentence. Absolute value is taken as the value of the negative sentiment polarity is going to be negative. As we are summarizing reviews the sentences having higher absolute polarity values are obviously better.

7. Numerical Words - The proportion of the numerical words in the total words is calculated and accordingly the score is assigned to the sentence. More the numerical words more better is the sentence considered.

8. Density of sentence - The number of key phrases is calculated and the score assigned is directly proportional to the number of key phrases found in the sentence.

III. Result

A. Timing Analysis

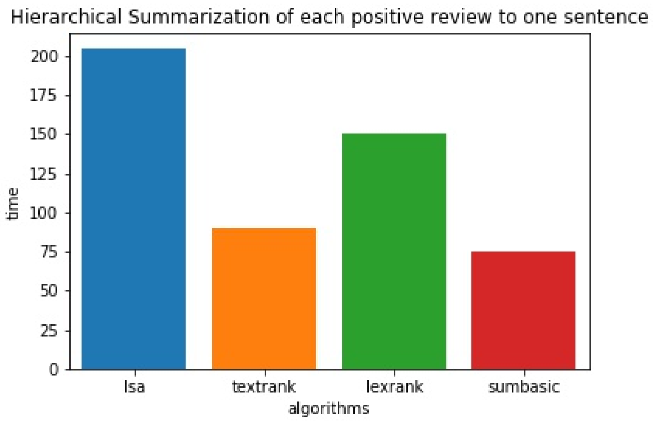


Fig. 9. Time (in milliseconds) Taken for Review Summarization

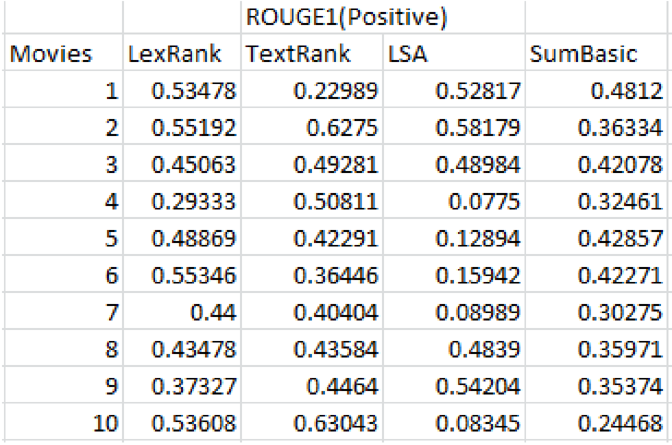
The above figure shows the timing profile of each of the four summarization algorithms incorporated in our model. SumBasic, one of the most widely used summarisation technique takes the least time while LSA takes the longest because of the matrix calculus associated with it.

B. Comparison of Summaries Using Rouge Score

We have used ROUGE score as a metric to compare the summaries generated using the four algorithms ie textrank, lexrank, LSA and sumbasic with the gold summary generated by using the sentences ranked on the different features mentioned above. The results obtained after the comparison of positive and negative summaries with the gold summary are summarized in the tables below. Average F-Measure scores have been shown in the table as being the harmonic mean of precision and recall they represent both. Movies 1,2,3.. represent the top ten movies.

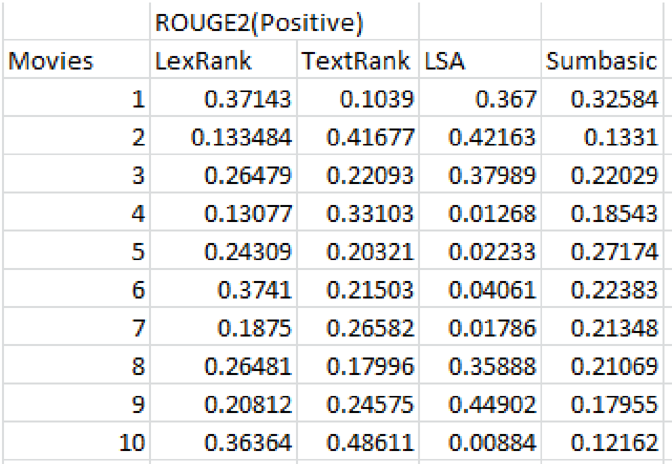
After analyzing the values it can be be concluded that LSA despite taking the longest time is giving the least accurate summary in most of the cases. Textrank turns out to be the most accurate algorithm over all the cases that is ROUGE-1 and ROUGE-2 score values for positive and negative summaries. Another observation that can be made is that the cases where the number of summaries is comparatively more (in the summarization of positive reviews of movies) lexrank and textrank are functioning better but the cases where the number of summaries is less (in the summarization of negative reviews of the movies) the sumbasic algorithm is also performing quite well.

So, the accuracy of algorithms is depending on the content to be summarized also.



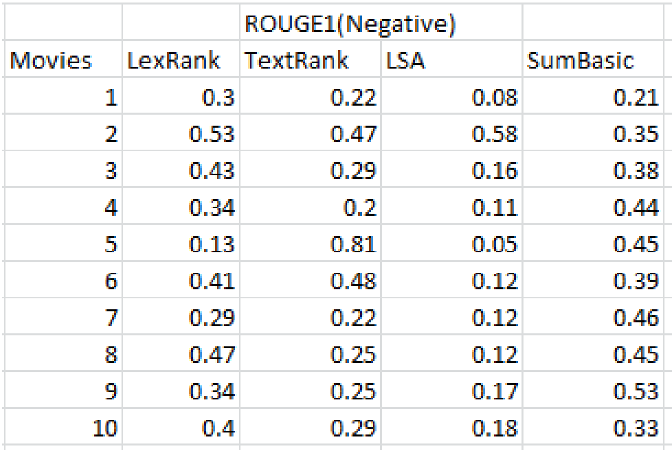
		ROUGE1(Positive)		
Movies	LexRank	TextRank	LSA	SumBasic
1	0.53478	0.22989	0.52817	0.4812
2	0.55192	0.6275	0.58179	0.36334
3	0.45063	0.49281	0.48984	0.42078
4	0.29333	0.50811	0.0775	0.32461
5	0.48869	0.42291	0.12894	0.42857
6	0.55346	0.36446	0.15942	0.42271
7	0.44	0.40404	0.08989	0.30275
8	0.43478	0.43584	0.4839	0.35971
9	0.37327	0.4464	0.54204	0.35374
10	0.53608	0.63043	0.08345	0.24468

Fig. 10. ROUGE-1 Scores for positive Summaries



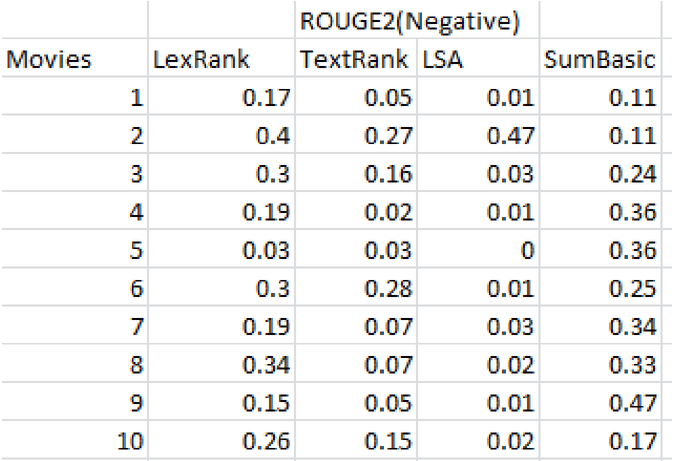
	ROUGE2(Positive)			
Movies	LexRank	TextRank	LSA	Sumbasic
1	0.37143	0.1039	0.367	0.32584
2	0.133484	0.41677	0.42163	0.1331
3	0.26479	0.22093	0.37989	0.22029
4	0.13077	0.33103	0.01268	0.18543
5	0.24309	0.20321	0.02233	0.27174
6	0.3741	0.21503	0.04061	0.22383
7	0.1875	0.26582	0.01786	0.21348
8	0.26481	0.17996	0.35888	0.21069
9	0.20812	0.24575	0.44902	0.17955
10	0.36364	0.48611	0.00884	0.12162

Fig. 11. ROUGE-2 Scores for positive Summaries



		ROUGE1(Negative)		
Movies	LexRank	TextRank	LSA	SumBasic
1	0.3	0.22	0.08	0.21
2	0.53	0.47	0.58	0.35
3	0.43	0.29	0.16	0.38
4	0.34	0.2	0.11	0.44
5	0.13	0.81	0.05	0.45
6	0.41	0.48	0.12	0.39
7	0.29	0.22	0.12	0.46
8	0.47	0.25	0.12	0.45
9	0.34	0.25	0.17	0.53
10	0.4	0.29	0.18	0.33

Fig. 12. ROUGE-1 Scores for Negative Summaries



		ROUGE2(Negative)		
Movies	LexRank	TextRank	LSA	SumBasic
1	0.17	0.05	0.01	0.11
2	0.4	0.27	0.47	0.11
3	0.3	0.16	0.03	0.24
4	0.19	0.02	0.01	0.36
5	0.03	0.03	0	0.36
6	0.3	0.28	0.01	0.25
7	0.19	0.07	0.03	0.34
8	0.34	0.07	0.02	0.33
9	0.15	0.05	0.01	0.47
10	0.26	0.15	0.02	0.17

Fig. 13. ROUGE-2 Scores for Negative Summaries

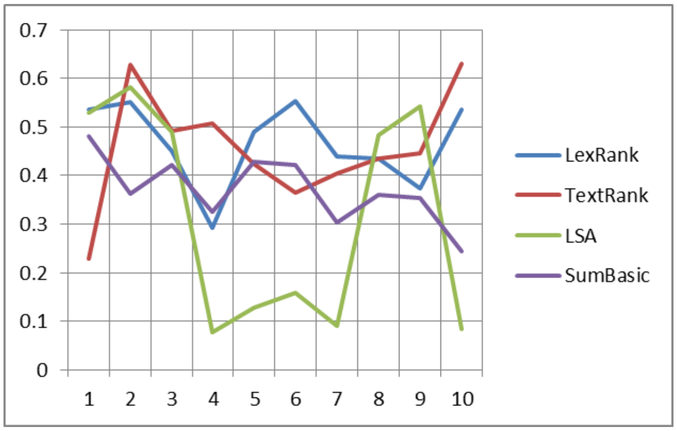


Fig. 14. Graph for ROUGE-1 Score for positive Summaries with Movies on x axis and Average F-measure values on Y axis

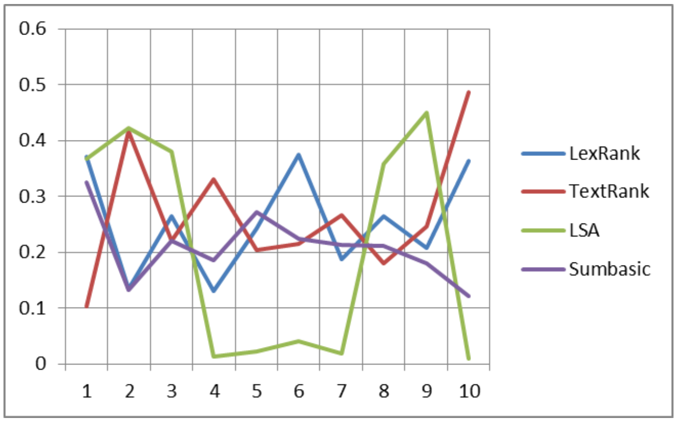


Fig. 15. Graph for ROUGE-2 Score for positive Summaries with Movies on x axis and Average F-measure values on Y axis

References

[1] G. Erkan and R. Radev "LexRank: Graph Based lexical Centrality as Saliance in Text Summarization," University of Michigan, Ann Harbor, MI 48109 USA

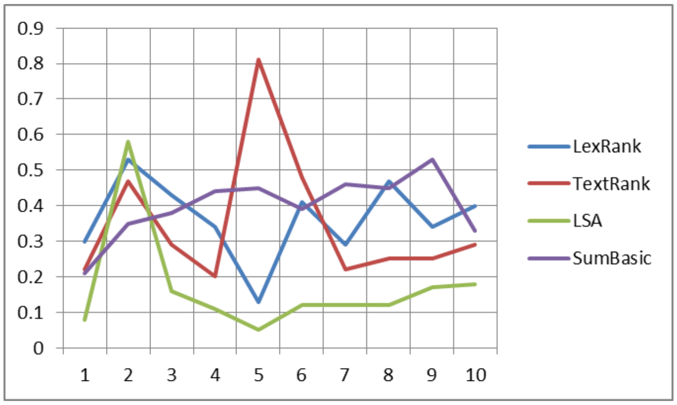


Fig. 16. Graph for ROUGE-1 Score for negative Summaries with Movies on x axis and Average F-measure values on Y axis

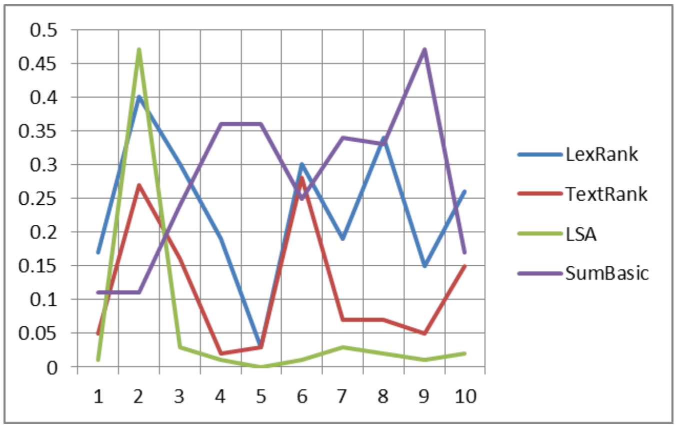


Fig. 17. Graph for ROUGE-2 Score for negative Summaries with Movies on x axis and Average F-measure values on Y axis

[2] Rada Mihalcea and Paul Tarau , TextRank : Bringing Order into Texts, Department of Computer Science, University of North Texas.

[3] Thomas K Landauer and Peter W. Foltz, "An introduction to Latent Semantic Analysis" Discourse Processes, 25, 259-284

[4] Lucy Vanderwende and Hisami Suzuki, "Beyond Sum Basic: Task focus Summarization with sentence Simplification and Lexical Expansion," Information Processing=g and Management, International Journal, Pages 1606-1618, Vol 43.

[5] DUC 2002 "Document understanding Conference 2002" www-nlpir.nist.gov/projects/duc/.

IV. Appendix

A. *Names of Top 10 Movies*

The names of the top 10 movies which are selected are as follows-

1. Star Wars Trilogy THX Digitally Mastered Edition
2. Star Wars - Episode I, The Phantom Menace VHS
3. The Lord of the Rings: The Fellowship of the Ring.
4. Firefly: The Complete Series
5. Marvel's: The Avengers
6. Avatar
7. The Hunger Games
8. The Hobbit: An Unexpected Journey
9. Prometheus

10. Star Trek Into Darkness

These movies were selected due to the high number of their reviews so that there is more amount of data to summarize.

B. Tables

The following are the tables using which bar graphs shown above have been drawn(corresponding to Fig. 1, Fig. 2, Fig 3).

This table contains the movieid (asin) and and the corresponding number of reviews for the top 10 movies selected.

Mean No of reviews per movie		
S.No.	**asin**	**Count**
1	0793906091	949
2	630575067X	901
3	B00003CWT6	1022
4	B0000AQS0F	928
5	B001KVZ6HK	1058
6	B002VPE1AW	1004
7	B003EYVXV4	1105
8	B0059XTU1S	933
9	B005LAIHXQ	1081
10	B009934S5M	1107

This table contains the mean number of words per review per movie.

Mean No of words per review in movie		
S.No.	**asin**	**Avg_Words**
1	0793906091	264.914647
2	630575067X	264.023307
3	B00003CWT6	268.896282
4	B0000AQS0F	181.544181
5	B001KVZ6HK	152.021739
6	B002VPE1AW	251.461155
7	B003EYVXV4	160.954751
8	B0059XTU1S	187.561629
9	B005LAIHXQ	219.283996
10	B009934S5M	176.707317

This table contains the mean number of sentences per review.

Mean No of sentences per review		
S.No.	**asin**	**Avg_Sentences**
1	0793906091	17.154900
2	630575067X	18.592675
3	B00003CWT6	17.499022
4	B0000AQS0F	12.731681
5	B001KVZ6HK	11.573724
6	B002VPE1AW	16.139442
7	B003EYVXV4	11.104977
8	B0059XTU1S	12.378349
9	B005LAIHXQ	14.414431
10	B009934S5M	12.335140