Movie Reviews Summarization

Hesham Nawaz, Ashhad Alam, Helen Lu, Ashar Farooq

Abstract

Multi-document summarization is a natural language processing task that involves creating a concise and coherent summary of multiple documents on a given topic. In the context of movie reviews, multi-document summarization can be used to automatically generate a summary from a large number of reviews for a particular film. This summary can then be used by individuals to quickly and easily understand the general sentiment and opinions of the reviewers, thus helping individuals make informed decisions about which movies they decide to watch. In this paper, we propose an adapted approach for multi-document summarization of movie reviews using a combination of pre-processing, sentiment analysis, and natural language processing techniques in addition to machine learning algorithms. We evaluate our approach on a Rotten Tomatoes dataset of movie reviews and show that it is effective at generating informative summaries with justified performance metrics.

1 Introduction

1.1 Motivation

The ability to automatically generate summaries of large collections of documents is becoming increasingly important in today's information-rich world. As the amount of available data continues to grow, it is difficult for individuals to make sense of this volume of information, such as thousands of movie reviews. Single-document summarization is insufficient in this context as each movie review is a distinct entity, reflecting somewhat independently generated perspectives on the movie. To capture the diversity of opinions and generate a coherent summary, we must thus rely on multi-document summarization techniques.

As many moviegoers factor into their decision to watch a movie what film critics have said about it, providing them a holistic overview of a particular movie by aggregating its reviews together can be of great help to them. In addition to its utility for individuals, this can also be valuable for organizations. For example, movie studios and production companies can use multi-document summarization to quickly and easily understand the general sentiment and opinions of reviewers for their films, helping them to make more informed decisions about their future projects as well as guide marketing and promotional efforts.

Overall, the motivation for this research is to develop approaches for multi-document summarization of movie reviews that are effective at generating informative summaries. By doing so, we hope to provide individuals and organizations with a useful tool for one of the most common recreational activities: movie watching.

1.2 Problem

The problem we aim to address in this research paper is the difficulty individuals and organizations face when trying to understand the overall view of a particular movie from many different movie reviews. Existing methods struggle to alleviate this problem - finding labels for what is considered a reasonable and informative summary in a dataset is challenging given the large amount of movies and how many reviews each of them has. Furthermore, summarizing movie reviews is also inherently hard because there are numerous sentiments presented by many reviewers, which are almost all subjective. As a result, a successful solution to this problem requires adapting existing techniques for multi-document summarization by specifically designing them to handle the challenges of summarizing movie reviews.

1.3 Dataset

The dataset we use is from Kaggle and it contains movies, critic reviews, and critically a reference summary written by a human, taking into account all of the movie reviews for a given movie. The first table contains approximately 17,700 movies with basic information for each movie, such as description, genre, etc. The second table contains roughly 1,100,300 movie reviews that can then be linked to the movies table via a unique id.

1.4 Preprocessing

Given the entire dataset, we remove all movies that do not have any reviews and all movies that do not have the human-generated reference summary. Furthermore, we remove the insignificant stop words in addition to movie reviews containing fewer than 100 characters.

1.5 Evaluation Metric

We use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) Metric to interpret the effectiveness of our approaches (Lin, 2004). Specifically, we calculate the F_1 scores of ROUGE-1 and ROUGE-2. ROUGE-1 refers to the overlap of unigrams between the produced and reference summaries while ROUGE-2 refers to the overlap of bigrams between the two summaries.

1.6 Code

The implementation of the approaches is primarily done in Python and the relevant libraries for data analysis and natural language processing. The full code repository is publicly accessible here

2 Related Works

This report builds on a considerable body of work that was consulted over the course of this research. Much of the past work in the domain of multi-document summarization has focused on product review summarization (Pawar et al., 2017) (Ly et al., 2011) but there has also been some work directed towards movies specifically (Khan et al., 2020).

A number of different methods have been used in these previous papers. In particular, Khan et al., 2020 (Khan et al., 2020) uses supervised learning and graph-based ranking algorithms applied on bag of words representations of the data. This is done in conjunction with a graphical optimization approach. Another relevant paper, Ly et al., 2011 (Ly et al., 2011) utilizes a combination of characteristic identification, clustering and latent semantic analysis.

There is also a great deal of research in natural language processing more generally that is also closely related to our work. Specifically, concepts such as efficient word representations (Mikolov et al., 2013), Jaccard similarity (Niwattanakul et al., 2013) and lexical centrality (Erkan and Radev, 2004) were integral to this paper.

Our methodology extends some of these previous approaches to a novel application, modifying the architecture to make it more suitable for the problem at hand, as well as incorporating more state-of-the-art techniques for some of the subtasks in the pipeline.

3 Baseline Approaches

3.1 Append-Summarize

3.1.1 Approach

This approach consists of appending together all the movie reviews for a particular movie and then running various summarization engines in order to produce the candidate summary for all the movie review documents. We want to take into account the perspectives of each movie review, thus appending all the movie reviews gives us the most holistic review. However, this review would be too long, thus summarization can result in a decent baseline generated summary.

We used 4 different summarization mechanisms as an experiment to see if a particular method resulted in better output of a generated summary. Refer to the text of all the movie reviews concatenated together for a given movie as appendedReviews.

We run two graph algorithms, TextRank and LexRank, on appendedReviews in order to generate the summary (Mihalcea and Tarau, 2004; Erkan and Radev, 2004). These algorithms are reasonable choices since they attempt to rank sentences by their importance to the meaning of appendedReviews, thus producing an informative summary. Typically, the similarity between sentences serve as the weight of the graph and the nodes can be the sentences of appendedReviews. In a similar vein of ranking sentences, we also ran NLTK's summarization package that relied on this calculating frequency and ranking sentences approach.

Another technique we used was a Latent Semantic Analysis (LSA) summarizer that utilized frequency techniques with singular value decomposition to generate a summary (Steinberger et al., 2004). LSA is a sensible approach for summarizing text because it is able to capture the relationships between the words in a document, allowing it to

Technique	ROUGE-1	ROUGE-2
TextRank	0.058	0.016
LexRank	0.071	0.018
LSA	0.101	0.019
NLTK	0.080	0.020

Table 1: The ROUGE-1 and ROUGE-2 F_1 scores are displayed for various techniques. These metrics measure the overlap between unigrams or bigrams between the generated summaries and the human-created reference summaries. The highest possible F_1 scores are 1.0, indicating perfect recall and precision. As expected, these values are not within a perfect system, thus a discrepancy exists.

effectively identify the key ideas and opinions expressed in the text.

3.1.2 Experiments and Results

To evaluate the effectiveness of this baseline approach, we conducted experiments via the 4 different summarization techniques of appendedReviews for about 1000 movies.

Table 1 shows the F_1 scores for ROUGE-1 and ROUGE-2 for the generated summaries across the different techniques. As can be derived from the table, the average F_1 scores of the 4 techniques for this baseline is 0.078 for ROUGE-1 and 0.018 for ROUGE-2. The low scores reflect generally poor results. However, this result is still somewhat unsurprising since the baseline approach overall involved appending together all of the movie reviews first and then running one of these techniques. This can be potentially problematic since there is a large amount of content that we are attempting to summarize concisely. Consequently, the sentiments of each movie review document are hard to pinpoint in the generated summary since there is high repetition in appendedReviews. In addition, we lose some context due to reordering of words as the placement of sentences of a text has been reconfigured, thus getting at the meaning of appendedReviews can be hard. We can note that the ROUGE-1 F_1 score for LSA is the highest among the scores, which can likely be explained due to the fact that we are capturing each word's relationship to each other word's relationship.

We can further explore the reasoning behind the generally low F_1 scores by computing the precision and recall scores for ROUGE-1 and ROUGE-2. These metrics can be seen in Figure 1 (ROUGE-1 scores) and Figure 2 (ROUGE-2 scores) for the NLTK summarization technique of appendedReviews. We note that in both cases, the precision values are fairly high while the recall values are fairly low across the 1000 movies. Precision measures what fraction of words that appear in our generated summary were also present in the reference summary. Recall, on the other hand, measures the opposite - what fraction of words from the reference summary were also present in the generated summary. Thus we can see from the graphs that the baseline model is relatively good at outputting only those words that are also in the reference summary but is not as successful at identifying all the words that were in the reference summary. This low recall drives down the F_1 scores.

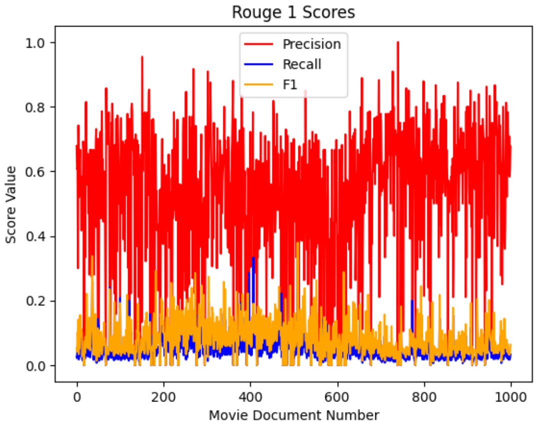

Figure 1: The F_1 scores for ROUGE-1 are generally driven down by the low recall across the 1000 movies, hence we do not see high F_1 scores (not close to 1.0).

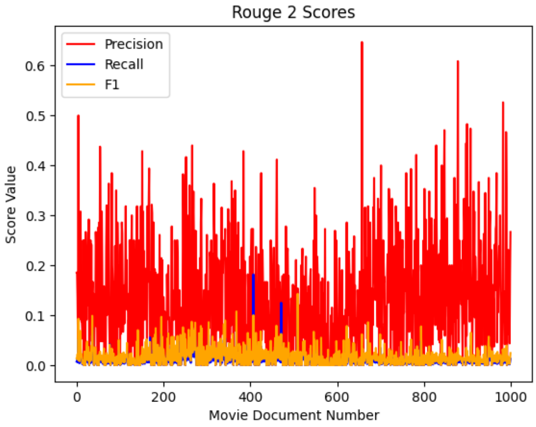

Figure 2: The precision scores for ROUGE-2 are generally high across the 1000 movies, although not perfect scores of 1.0 since the summaries are ultimately predictions.

These results can be generalized with this type of approach as we tried it across multiple different techniques of text summarization. Despite the fairly low scores, the results can still be trusted because we used standard and proven algorithms via several Python packages like sumy in order to abstract away complicated details, thus ensuring reliability.

3.2 Similarity-Pick-One

3.2.1 Approach

This approach consists of computing Jaccard similarity scores between all pairs of movie review documents (Niwattanakul et al., 2013). This enables us to find the movie review that is the most similar to every other review for a given movie. The higher the Jaccard similarity score is, the more two documents are similar. This is a reasonable approach since we know from other applications that neighbors, surroundings, similarities, and connections to other entities showcases a relationship that can be used to infer information. In other words, we can pick the movie review that seems seems most similar to other reviews in order to capture the essence of many different movie reviews for a given movie.

3.2.2 Experiments and Results

This baseline approach can also be analyzed through ROUGE metric scores. After performing the calculations, we can note that the average F_1 score for ROUGE-1 is about 0.204 while the average F_1 score for ROUGE-2 is about 0.057. These values are significantly better than all of the baseline approaches that we attempted. This is partly due to the fact that we can take into account information and meaning from many documents at the same time since we want a movie document that is similar to all the other movie documents.

Decomposing the F_1 scores into precision and recall values, such as in Figures 3 and 4, can help illuminate why the F_1 scores are higher. As can be seen in both cases, the recall score is much higher and the precision score is much lower across the 1000 movies. This means that this model is relatively bad at outputting only those words that are also in the reference summary but is much more successful at identifying all the words that were in the reference summary. The higher recall score is driving up the F_1 scores in this case.

This result can generalize since we noticed that finding a representative of some entity like a sentence or paragraph could result in reasonable metrics, especially compared to the initial baseline. This representative entity captures the meaning of many movie review documents. The main takeaway for this baseline approach is the potential of diving into other similarity-based techniques. The results should be trusted since we are using mathematical quantifications of what constitutes similar sentences.

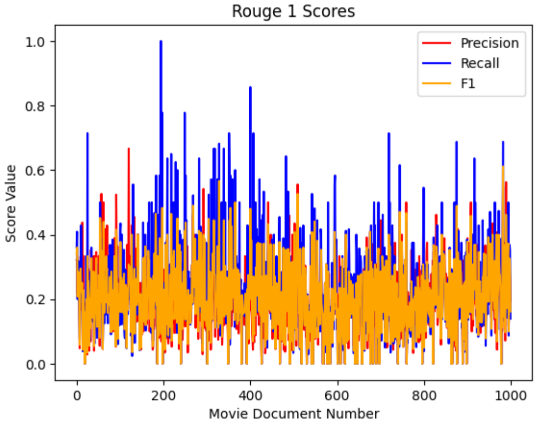

Figure 3: The F_1 scores for ROUGE-1 are generally higher compared to other baseline approaches as the recall scores are elevated across the 1000 movies.

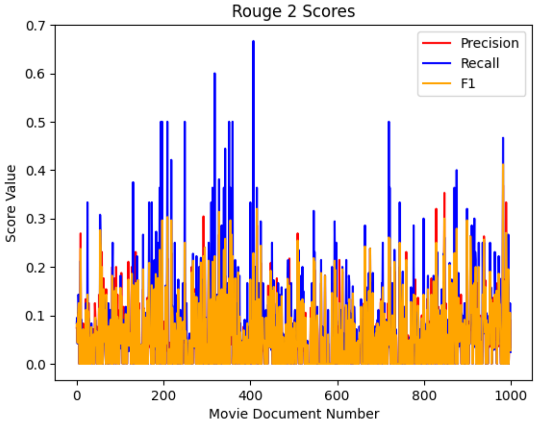

Figure 4: The precision scores for ROUGE-2 are generally lower as compared to other baseline approaches.

4 Approaches

4.1 Primary Approach - Summarization by Characteristics

4.1.1 Approach

This approach consists of identifying key characteristics that relate to a given movie and then choosing a positive and negative representative sentence for each characteristic if they exist based on the paper by Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin and Min-Yen Kan (Ly et al., 2011). Hence, the output summary of each movie consists of a list of characteristics and representative sentences that relate to each characteristic. To achieve this, we have implemented a model that is split into two phases: characteristic identification and summarization. The overall architecture of this approach is depicted in the figure below.

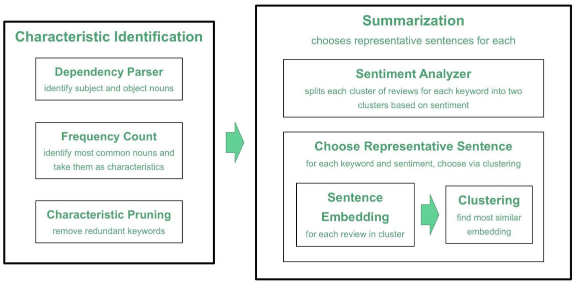
Characteristic Identification
Dependency Parser
identify subject and object nouns
Frequency Count
identify most common nouns and take them as characteristics
Characteristic Pruning
remove redundant keywords
Summarization
chooses representative sentences for each
Sentiment Analyzer
splits each cluster of reviews for each keyword into two clusters based on sentiment
Choose Representative Sentence
for each keyword and sentiment, choose via clustering
Sentence Embedding
for each review in cluster
Clustering
find most similar embedding

Figure 5: Model architecture for the summarization by characteristics approach.

Characteristic identification, as the name suggests, identifies key characteristics that appear in the reviews for a given movie that will be summarized on. This phase has three steps. We first used Stanford's POS Tagger to identify all nouns that play a subject or object role in the sentence and discard all other nouns. After obtaining a preliminary set of candidates to serve as characteristics, we calculate the support of each noun which we then use to identify the characteristics with highest support. With this refined list of candidates, we prune the characteristics one last time to remove redundant keywords. Specifically, we focus on removing keywords that only have a singular word if there exists a compound keyword which has a higher support.

Summarization chooses representative sentences for each characteristic identified by the previous phase. The first step is to apply a sentiment analyzer that splits the review space into positive and negative sentiment. Then for each sentiment, it creates clusters of reviews that contain the characteristic of interest. We then embed all reviews in the same cluster and find the review that is most similar to all embeddings in the cluster. That review would then be the representative review for the characteristic with the particular sentiment.

The premise of choosing this approach was that based on the results from the baseline, summaries comprised of representative sentences perform better. Moreover, in the context of movie summarization, individual characteristics such as effects, plot, etc. may be of more importance. As a result, we wanted to implement this characteristic-based divide and conquer approach to summarization.

4.1.2 Preprocessing

The preprocessing step was essentially the characteristic identification step where we used the Stanford POS Tagger, support filtering and characteristic pruning to narrow down the set of possible characteristics we filter on.

4.1.3 Results

Our model's F_1 scores for both ROUGE-1 and ROUGE-2 were not very good. The ROUGE-1 F_1 score was 0.064 while the ROUGE-2 F2 score is 0.016. This is partly expected since the output of our summaries are generally very long because it produces a long summary consisting of multiple representative sentences (one per characteristic), which are often a lot longer than the critic summaries that we used as our ground truths. Despite its poor F_1 performance, we still believe that our approach addresses a different, more informative form of summarization that is also of great value.

4.2 Alternative Approach - Weighted Graph Rank Algorithm

4.2.1 Approach

This approach consists of classifying reviews as positive or negative, and picking the most representative sentences within using a graph based ranking algorithm (Khan et al., 2020). This allows us to return a smaller subset of sentences that capture the essence of all of the reviews.

To do this, we will first classify the reviews into "Fresh" (positive reviews) or "Rotten" (negative reviews) using Naive Bayes' classifier (Leung, 2007). Then, we will get a vector representation of every sentence within each review using word2vec (Mikolov et al., 2013), and take the mean of all the words in the sentence to be the sentence representative vector.

We will now calculate the pairwise cosine similarity of the sentences. Using this, we will create a graph, where each sentence is represented by a node in the graph, and an edge exists between two sentences if the cosine similarity meets a certain threshold. We will use a lower bound of 0, as we want to make sure the reviews have some similarity if we're connecting them. In addition, we will use 0.5 as the upper bound, so that we are not returning reviews that say essentially the same thing multiple times. The weight of each edge will be the similarity score between the two sentences.

We will use these nodes and edges calculate the weighted graph rank (WGR) of each node using the following formula (note that this is very similar to the pagerank (Xing and Ghorbani, 2004) formula):

$$WGR(v_i) = 1-d+d* \sum_{(v_j,v_i)\in E} \left(\frac{WGR(v_j) * w_{ji}}{\sum_{(v_j,v_k)\in E} w_{jk}} \right),$$

where, d is the damping factor (which is generally set to 0.85), E is the set of all edges, and w_{ij} is the weight of the edge from v_i to v_j. Once we have run a few iterations of this algorithm, or if the weighted graph ranks start to converge, we can simply pick the 10 sentences that have the highest rank, and use them as our summary.

4.2.2 Preprocessing

The preprocessing of the data first of all involved converting the category of reviews ("Fresh" or "Rotten") to 1s and 0s. After that, we needed to split each review into sentences, and apply word2vec embedding on each of these sentences.

4.2.3 Experiments and Results

This paper had a large number of parameters that could be experimented with. Some of these include the damping factor, and the different thresholds for when to create an edge between the different sentence nodes. However, this entire process is also quite computationally expensive. For instance, if for a given movie, we have 100 reviews (best case, each review is exactly 1 sentence), that would mean that we are working with a 100x100 matrix, and updating that matrix for each iteration of the graph rank algorithm requires a considerable amount of time. Consequently, the process of generating the complete matrix necessary for this approach proved computationally intractable and so, we do not have any experiments or results yet to present for this approach.

5 Summary and Future Improvements

The approaches we have considered in this paper, while clearly imperfect, show great promise for quickly gaining a broad understanding of the opinions of movie critics and audiences. In particular, summarization by characteristic seems to be a very intuitive and practical way to address this challenge - this summary can provide a useful overview of the overall sentiment of the reviews, as well as the most commonly mentioned keywords and themes.

In the future, there are several improvements that can be made to the multi-document summarization of movie reviews. The main areas are characteristic generation, sentiment analyzer and potentially a different similarity score. Utilizing the support, which is essentially a frequency count, as a mechanism to select the characteristics might not have been the best measure of keywords. Additionally, based on the reviews, it seems that the characteristics that we want to summarize on are not always directly in the text itself. This calls for an alternative to generate characteristics to summarize on.

A better approach for characteristic identification would be to use a combination of automatic and manual keyword generation. This would involve using natural language processing techniques to identify the most important words in the reviews, and then manually reviewing and adjusting the list of keywords to ensure that they accurately reflect the content of the reviews. Another potential approach is to remove the characteritic identification phase as a whole. Instead, we could provide a fixed list of characteristics to summarize on, create embeddings on the reviews and cluster all reviews similar to the characteristics and then generate summaries from there. A more manual approach is to first cluster all reviews and then manually check it see if there is any pattern or notable characteristics discussed for each cluster.

Another area of improvement is to use a better sentiment analyzer. Currently, the implementation utilizes the NLTK sentiment analyzer, which we found to misclassify reviews because it only looks at the words rather than the grammatical structure of the review, which may change how the negative words affect the sentiment of the sentence. Since our clusters are partially based on the sentiment, it is worth finding alternatives for a more sophisticated sentiment analysis tool that can more accurately determine the sentiment of a given review. This would help the system to more accurately summarize the overall sentiment of the reviews.

Finally, the system could be improved by experimenting with different similarity scores and fine-tuning the sentence embeddings. Currently, the system uses a fixed similarity score to determine which sentences are most representative of the overall content of the reviews. However, by experimenting with different similarity scores and fine-tuning the sentence embeddings, it may be possible to improve the quality of the representative sentences that are chosen. This would help to ensure that the summary accurately reflects the content of the reviews.

6 Impact Statement

The datasets used in this paper are available publicly and all research and processing work was conducted solely by the authors of this paper. One potential ethical concern with the use of multi-document summarization of movie reviews is that it could lead to the amplification of biased or skewed opinions. If a summarization algorithm is trained on a large dataset of movie reviews that is not representative of the overall population's opinions, it may produce summaries that favor certain perspectives and ignore others. This could result in a distorted understanding of a movie's quality, and could even potentially harm the reputation of a film if the summary is widely shared.

Another potential societal concern is that the use of multi-document summarization in this context could lead to a decrease in critical thinking and analysis among moviegoers. If people rely solely on summaries of reviews to determine whether they should watch a movie, they may not fully engage with the original reviews and form their own opinions. This could result in a loss of individual perspectives and a homogenization of opinions, which could ultimately be detrimental to the diversity and richness of the movie-going experience. This may also present an additional concern for the reviewers whose jobs may be impacted by the development of such a technology.

7 Contributions

Ashar worked on generating baseline results for the Append-Summarize approach. Hesham worked on generating baselines results for the Similarity-Pick-One approach. Helen worked on creating a model using the Characteristic Summarization approach. Ashhad worked on the weighted graph rank approach.

Acknowledgements

The authors would like to thank the following people for their support and contributions to this research:

First and foremost, we would like to express our gratitude to the course instructors, Yoon Kim and Chris Tanner, for their invaluable guidance, support, and lectures throughout the class and research process.

We would also like to thank Rotten Tomatoes for providing us with access to their extensive movie review dataset, which served as the basis for our experiments.

Finally, we would like to acknowledge the support of Faraz Faruqi for providing valuable feedback on our research and for his insightful suggestions for approaches.

We are grateful to all of the above for their contributions to our research.

References

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Atif Khan, Muhammad Adnan Gul, Mahdi Zareei, RR Biswal, Asim Zeb, Muhammad Naeem, Yousaf Saeed, and Naomie Salim. 2020. Movie review summarization using supervised learning and graph-based ranking algorithm. *Computational intelligence and neuroscience*, 2020.

K Ming Leung. 2007. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007:123–156.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin, and Min-Yen Kan. 2011. Product review summarization from a deeper perspective. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, page 311–314, New York, NY, USA. Association for Computing Machinery.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

Priya Pawar, Siddhesha Tandel, Shweta Bore, and Nikita Patil. 2017. Online product review summarization. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–3.

Josef Steinberger, Karel Jezek, et al. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4(93-100):8.

Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, pages 305–314. IEEE.