International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1077-1086

Using User-Given Scores in the Summarization ofPosts and Replies in Online Forums

Prakash C O, Somak Mukherjee, Swati Maruvalli
Department of Computer Science and Engineering
PES University
Bangalore, India

Abstract—*We develop a tool for the use of individuals who are looking to condense and parse through large amounts of information available online in forums as a part of topic-based information extraction or to ease the understand-ability of available information. Using this tool, AAAA, users will be able to more efficiently access information that may otherwise be difficult to identify or isolate by specifying parameters of information retrieval. We build upon basic TextRank to include user scores in the calculation of importance of forum posts using valuable available human input. This is then augmented via topic-modeling for further improvements in usability and accuracy. This project involves the design and implementation of the tool, AAAA, the scope of which includes the functionality for topic-based selection of posts, the training system for identifying user scoring trends and the final piece of software which utilizes these elements to provide an easy and effective means of information gathering to the user.*

Keywords—*information, parameters, topic-modeling, score*

Date of Submission: 05-06-2022 Date of acceptance: 20-06-2022

I. INTRODUCTION

Forum threads and online conversations contain valuable information for human readers. A large amount of highly useful information is nowadays available not as set papers or even dedicated blog posts, but as part of a discussion between multiple individuals in an online forum thread. These discussions cover a wide variety of technical, creative an opinion-based topics that contain multiple snippets of useful specific information.

However, due to the relatively unstructured and informalnature of the forum threads on which these discussions occur, the relevant information is often difficult to parse for end users who are looking for information regarding a certain element of a specific discussion topic. Since almost anyone can post, there is a lot of informational noise present. Posts that do not add relevant information are common, such as posts that simply offer thanks and congratulations, or those that agree or disagree without saying anything of substance. Posts in more general threads will have information that is not relevant to the user at the moment since topics of discussion will diverge as time goes on. In a thread discussing a particular module for an application, for example, installation, debugging, techniques are all validtopics of discussion, but are not useful to all users and are not useful all the time. When they are not what the user is looking for, they will only serve in distracting the user and slowing them down. Thus, a need arises to be able to effectively and accurately summarize forum threads, filteringout useless information, identifying topics of discussion and accurately doing so in accordance with human usefulness.

Summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information withinthe original content. Multi-document summarization creates information reports that are both concise and comprehensive. With different opinions being put together & outlined, every topic is described from multiple perspectives within a single document. While the goal of a brief summary is to simplify information search and cut the time by pointing to the most relevant source documents, comprehensive multi-document summary should itself contain the required information, hence limiting the need for accessing original files to cases when refinement is required. Automatic summaries present information extracted from multiple sources algorithmically, without any editorial touch or subjective human intervention,thus making it completely unbiased.

Summarization is usually carried out in two distinct styles. Extractive summarization means identifying importantsections of the text and generating them verbatim producing a subset of the sentences from the original text; while abstractive summarization reproduces important material in a new way after interpretation and examination of the text using advanced natural language techniques to generate a new shorter text that conveys the most critical information from the original one. Obviously, abstractive summarization is more advanced and

closer to human-like interpretation. Though it has more potential (and is generally more interesting for researchers and developers), so far the more traditional methods have proved to yield better results.

Forums are a special subset of the summarization problem since they are not fully represented by single document, and instead need to be modelled as dialogues between multiple parties. And since we wish to present relevant information in the form it was originally presented, we will be implementing extractive summarization.

In addition to performing extraction, some basic topic modelling will be performed to identify dominant topics of discussion present in the thread to more easily access information that is relevant to users.

Topic modelling is a machine learning technique that automatically analyses text data to determine cluster words for a set of documents. This is known as 'unsupervised' machine learning because it doesn't require a predefined list of tags or training data that's been previously classified by humans. Topic modelling involves counting words and grouping similar word patterns to infer topics within unstructured data. By detecting patterns such as word frequency and distance between words, a topic model clusters feedback that is similar, and words and expressions that appear most often.

For the purpose of topic modelling a large constantly varying set of discussion topics across multiple domains, weuse Latent Dirichlet Allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Each topic is, in turn, modelled as an infinite mixture over an underlying set of topic probabilities. In the context of text modelling, the topic probabilities provide an explicit representation of a document.

While previous work has been done in the field of summarizing forums, none appear to utilise forum specific information that is readily available, such as user given scores, in extraction algorithms. Human judgement is frequently used as a gold standard for training machine learning algorithms and we believe that including valuable user input such as forum post scores while accounting for human biases and scoring trends will significantly improve the data content of the final extracted information. This new method, taking advantage of a natural forum conversation structure and the topics modelled on the selected thread, willimprove result quality of extracted posts.

II. RELATED WORK

A. Survey on Extractive Text Summarization

Text summarization has become a crucial means for supporting and presenting text content in the context of the internet and the information age. It is time consuming for humans to manually summarize large disparate documents oftext. There is a huge amount of textual content available on the internet. But the internet contributes more data than is desired. Looking for appropriate documents through an awe-inspiring number of reports offered, and fascinating a high volume of important information. Automatic text summarization aims to compress the original text into aspecific version which preserves its informational contents. The main advantage of a text summarization is reading time of the user are often reduced. A good system for text summarization, while replicating the various theme of the document, should keep repetition to a minimum. Text Summarization methods are commonly classified as either abstractive and extractive summarization. Extractive summarization is the process of identifying valuable portionsof the input data and reusing the text as it was already presentby using a certain number of the sentences from the initial document. The significance of sentences is strongly supported statistical and linguistic features of sentences. This paper depicts the features for extractive text summarization, describes extractive text summarization methods, illustrates inferences made, represent challenges and future research directions, and detail about evaluation metrics. It provides a host of methods which can be followedduring development for extractive summarization.

B. Topic-Focused Summarization of Chat Conversations

In the past, the communication of users through social media has seen an exponential increase. A large portion of information transfer happens online in the form of forum posts and thread discussions. These discussions are often comprised of a significant number of users and a singlediscussion tends to cover a wide range of topics alongside minor irrelevant portions in between. In order to effectively understand the contents of these long and involved discussions, summarization becomes essential. The summaries provided can be of very good commercial and educational value, and can be used to analyse the impact of virtual social interactions and virtual organizational culture on product development. The approach as described in the paper consists of two phases. Topic modelling is done using web documents to find the main discussion topic in the chat. Following this, in the summarization phase, a semantic word space is built to score sentences based on their association with the primary topic.

C. TextRank: Bringing Order into Texts

In this paper, TextRank is introduced as a graph based ranking model for graphs extracted from natural language texts. TextRank is applied to two language processing tasks (unsupervised keyword extraction, sentence extraction), evaluated, and it is shown that the results obtained with TextRank are competitive with state-of-the-art systems developed in these areas.

Using graphs built on texts, TextRank identifies the connections between units within a text, and a system of recommendation is utilized which recommends other related text units, and the strength of the recommendation is calculated recursively. While identifying important sentences contained within a text, a sentence connects to another sentence that refer to similar concepts as being useful for the understanding of the text as a whole. The sentences that are highly recommended by other sentences within the text are more likely to be informative and are given a better score.

In this way, TextRank is capable of scoring text units based on the "importance" of other text units they link to instead of simple graph connectivity. The units selected by it for a given application are those most recommended byrelated text units within the text, with preference given to therecommendations made by most influential ones. Inside a proper sensible text fragment, related units tend to form a "Web" of connections that comes close to the model humans would create about a specific context with respect to the process of interpreting a discussion. No deep linguistic knowledge or domain specific annotated corpora is necessary, which makes it highly portable.

III. PROBLEM DEFINITION

The aim is to design an application which will assists user in efficiently gathering information from online forums. There are previously established methods on summarizing the information found in forums. Similarly, this problemfocuses on designing and implementing a tool which can utilize available information to allow users on to categorically extract relevant information.

A. Stages of implementation:

- Write out the overall design of the application.
- Write out the steps to augment TextRank algorithmby including user scores.
- Implement basic form of TextRank for ExtractiveSummarization.
- Implement basic form of LDA for Topic Modelling.
- Design and implement user score considerationwhen calculating rank of posts.
- Implement UI and unify functionality.

B. Functionality:

- Perform topic-wise summarization and post hidingfor better readability
- Allow user to dynamically show and hide posts asthey need
- Calculate importance of posts based on both textualrichness and user score (whenever available)
- Automatically gather complete thread data andgenerate summary on the fly when url is provided

IV. OUR APPROACH

The major steps in the process of designing the system ofextraction are as follows:

A. Data-set Creation

The data for training the score trends is extracted from the already available datasets for the different stackexchangedomains. The data is parsed from an xml that contains a large amount of data, organised into an intermediary format that better represents the structure of the posts within threadsand all relevant features.

Data points which are retained consist of:

- Post ID
- Post Text Content
- Post Score
- Time of creation of post
- No. of views for the Thread as a whole
- Type of Post

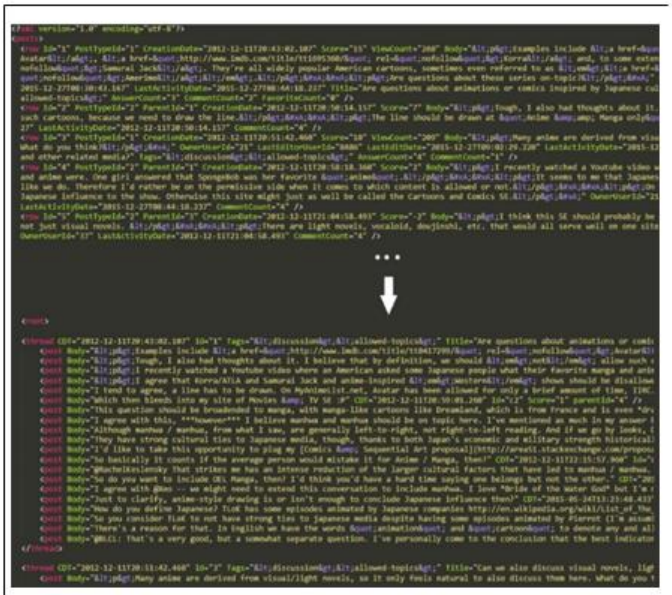

Figure 1: Restructuring available data

This intermediate step is then followed by further processing and clean-up in which the structural information is utilized to generate the necessary timing, traffic and proportional scoring data that will be utilized in score trend analysis.

Data points which are retained include:

- Type of post
- Difference in time from the creation of the original post that started the discussion thread and the current post
- Post Score
- Thread Views
- Post Score divided by Thread Views

Figure 2: Extracting necessary data

B. User Scores and Score Trend Analysis

The work that has been done regarding summarizingforum content that is publicly available disregards the user given scores that sites contain, in favor of increasinglycomplex text processing algorithms. We include user scores along with basic text rank to produce more representative outputs.

User scores provide necessary human insight on posts, that can be unclear to text processing algorithms. Human input is often considered a gold standard when training ML algorithms, so readily available human input should be usefulin determining output.

However certain precautions need to be taken when dealing with human given scores, which may end up misrepresenting the data further if not handled.

1) The Effect of Forum Traffic:

On most online forums, new posts are presented on a front page of sorts, sorted by order of newness, and gradually move further and further down the page until they vanish off it completely. A new post made in a thread may push it further up in certain circumstances, but in general, older threads get less and less traffic as time goes on.

As a result, threads get a lot of views near the beginning and gradually lower views as time passes. Thus posts made near the start get more exposure and have a higher chance of being upvoted. This can also be observed mathematically, since threads that get more views in general have higher scoring posts. An unfortunate side-effect of this phenomena is that a relevant, useful or better post made later on in the life cycle of a thread can have a lower score simply due to not enough people seeing it.

Thus, user score needs to be weighted based on time since post is made after the thread is created, and the overall number of views the thread receives. This is the hurdle which is overcome by Score Trend Analysis.

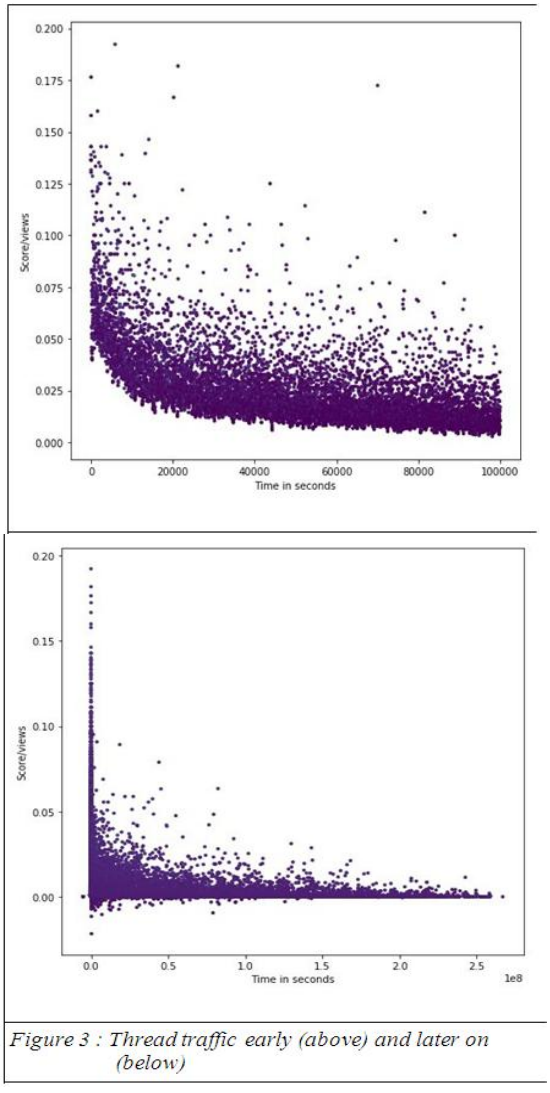

Figure 3 : Thread traffic early (above) and later on (below)

Fig 4: Effect of scores on controversial opinions

2) The Effect of Human Bias:

People in a forum tend to mass-downvote opinions unpopular in their circles. Certain opinions on topics, especially those of a political or religious nature, may be considered controversial in certain groups, which then leads to such posts gaining low or even negative scores, despite these posts often containing noteworthy, if not valuable information. In a similar vein, we cannot count on people to be completely unbiased, and certain circles may prefer certain radical opinions. Certain circles also upvote specific types of posts due to the culture of the forum, which can throw off the weighting. This can be observed with the"cake-day" upvotes that posts made across Reddit get where,even on serious subreddits, posts made by individuals on their birthdays get upvoted for that fact alone.

Thus, we need to include human input but it cannot be allowed to completely overwhelm the score generated from the text-rank.

The final result of the dataset from the Score Dataset Preparation Module is used in training the system to identifythe values that fit the pre-set equation of the curve to the score information dataset. These values indicate the rate of falloff of scores over time for the given domain of discussion. Therefore, once inverted, we will get a relative weight appropriate for the user score given to the post.

These values will be different for different groups of discussion. For websites with more isolated communities, each domain will have its own set of values that describe score falloff over time, whereas for websites with a more focused topic, or one with a fluid user base, such as Reddit, these values can be expected to be closer between domains, so values must be calculated for larger domain-groups or even sites as a whole.

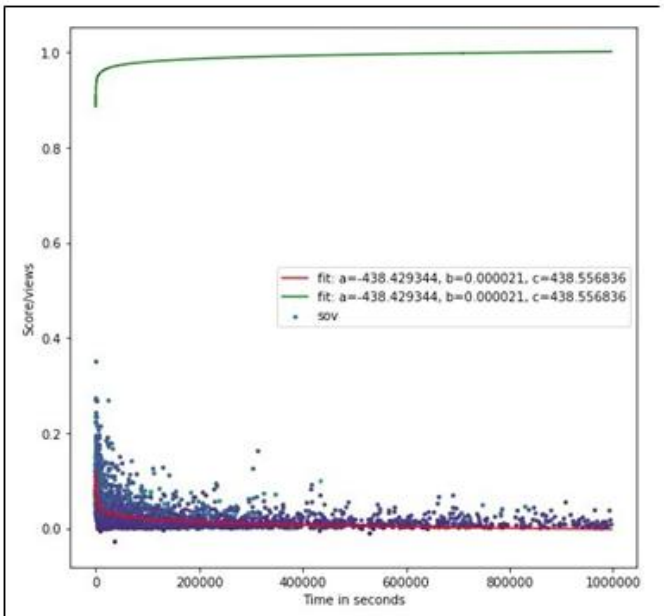

Figure 5: Curves fitted to Score trends

C. Scraping

The user will provide the url for the thread they desire summarized. This module will scrape said thread, given that it belongs to a supported website, and proceed to temporarily store the relevant information into a file that is efficiently usable by the rest of the system.

D. Topic Modeling:

To model the topics present in a given thread, we use LDA. There is no pre-training done, since topics vary wildly within threads and a comprehensive list of topics would be very large. Instead, unsupervised topic-modelling isperformed that will allow us to identify hidden semantics within the thread itself, which users will be able to view and understand the topics.

We consider each complete post as a unit of information. We start by performing some minor cleanup and returning a set of lowercase english tokens which are then reduced to the respective base words using the WordNet Lemmatiser. All stop-words are then filtered out. We are left with sets of tokens that represent each statement. This data is then converted into a dictionary and then a bag of words, on which the LDAModel available in gensim is trained. Both the model and the bag of words are saved for future use.

The final model is then applied on all the posts in the current thread, assigning each post a set of topics which they belong to. This data is saved to a local file for easy access during filtration later on.

The number of topics is decided upon by the user, and they can change this number at any time in the future. This will require a recomputation of the model and should in general be avoided since it may cause some slowdown, but since a lot of the processing is already performed and then saved locally, future computations will be faster.

E. Extractive Summarization:

1) Text Rank:

Extractive text summarization is all about finding the more important sentences from a document as a summary ofthat document. Our approach is using the TextRank algorithm to find these 'important' sentences. The storedthread is broken up into a set of posts. Thus we have a set of documents, which are further tokenized into a set of root words that represents the content of each post. This is then used to generate a term-document matrix.

This matrix is then used to create a graph where each node represents a complete post and an edge represents that they have shared tokens. The edge weight is the number of tokens that are shared by both of the posts. Ranks are then generated using the page-rank functionality. This gives us a basic TextRank score for each of the posts.

2) User Scores:

Once the basic TextRank is calculated, we include the influence of user scores. The formula used to properly weightthe influence the weight of various scores is:

$$\frac{(S1 * W1)+(S2 * W2a * W2b(time))}{(W1 + W2a)}$$

This is a weighted mean that weighs the TextRank Score S1 with weight W1, and weighs User Score S2 with constant weight W2a and variable weight W2b which is a function of time difference of the current post since the initial post in the thread was created.

The weight W1 is set to a default value of 1.

The weight W2a is present to prevent the user score fromoverwhelming the TextRank score completely, and can only be reached via manual opinion-based testing, in which the value is randomly increased and decreased by variable amounts, leading to a variety of outputs, from which testers select the one that best captures the available information. After iterating over this process multiple times, a specific

value of W2a can be reached for a given website. In effect, W2a exists to counteract the Effect of Human Bias when dealing with User Scores.

The function W2b(x) is defined as

$$1 - (ax^b + c)$$

Where x is the time difference of the current post since the initial post in the thread was created and a, b and c are obtained from Score Trend Analysis on training data for a relevant domain. The given equation of the curve for W2b(x) is used because it was observed to fit best with the distribution of scoring data during Score Trend Analysis.

Following this, a new set of ranks is calculated for each of the posts, one which better represents the input of the users who use the forums with completely overwhelming theinformation present in the textual contents of said post.

Once the ranks are generated, a threshold value for the score, represented by a deviation from the mean, which is provided by the user, is used to select a set of posts all of which have a score higher than or equal to the calculated threshold.

3) Handling Post Hierarchy:

In order to preserve context and the accurate representation of a given discussion, all the posts that are higher up in a hierarchy with respect to a given selected postis also included in the final summarization. This can be done easily since one of the features preserved during scraping is the unique id of any parent post, with the highest possible position in the hierarchy occupied by the original post that started the thread, since it can be assumed that most repliesin the thread will be in response to said post. Thus, all the posts which have been selected as apporpriate for the summary up to this point are returned as output.

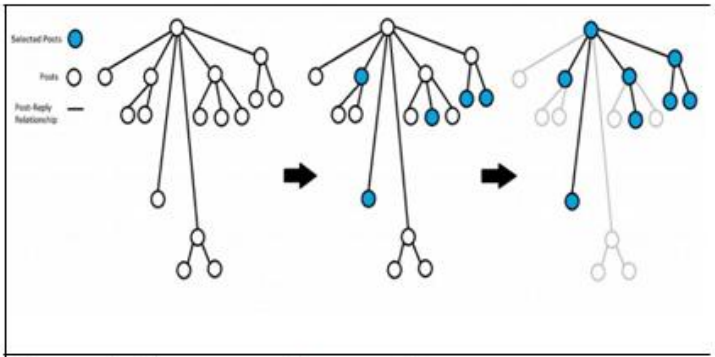

Figure 6: Recursive selection of Post parents

4) User Input at Runtime:

The user has certain parameters for extraction that theycan alter. This includes:

- Threshold value for scores during extraction.Formula used to calculate threshold is as follows:
  - threshold = sum(scores)/(no. of scores) + offset
  - The value offset is accepted from the user andmay be positive or negative.
  - A low value is more flexible and allows lowerscores, while higher values are stricter.
- Number of topics to be used during topic modelling
- Topics to be included or excluded during extractionbased on those which have been modelled. This is done using the locally stored topic assignments done during the topic modelling stage so that posts can be included and hidden at hight speed.

V. RESULTS

In our tests, the results produced by our algorithm were in general more comprehensive and upon review, seen as more representative of the original thread discussion than basic TextRank.

We found that the values for W2a that produced the mostrepresentative results during testing were very small (e.g. a value of 0.005 for the philosophy board of stackexchange). This is not surprising, since the idea was to only nudge the TextRank scores in a direction that reflected the human intentof the forum users. This value can be more effectively finalized by more extensive testing by more individuals. Since the current value exists by the testing input of only twoindividuals, it may be skewed towards our individual biases.

VI.DISCUSSION

We found that the values for W2a that produced the mostrepresentative results during testing were very small (e.g. a value of 0.005 for the philosophy board of stackexchange). This is not surprising, since the idea was to only nudge the TextRank scores in a direction that reflected the human intentof the forum users. This value can be more effectively finalised by more extensive testing by more individuals. Since the current value exists by the testing input of only twoindividuals, it may be skewed towards our individual biases.

Values of a, b, c used in the equation for W2b(x) are calculated based upon an arbitrary granularity of domains, which needs to be further controlled in future based on the content and culture of a given forum or website. For a website like reddit which encourages cross-posting across domains, it would make sense to generate these values for the entire website, or if we wish to be more specific, a set of sub-reddits that share common topics of discussion or discuss closely related ideas, since the voting patterns of the individuals who visit these forums will be largely similar. Onthe other hand, a website that does not explicitly encourage cross-posting and instead favours more contained, domain specific discussions, the values need to be calculated specificto said domains. Even higher specificity is possible by considering the types of users who post content, additional information such as tags or flairs that indicate the type of thepost being made, but care must be taken to not run into the problem of overfitting.

For the purposes of this project, we relied heavily on freely available stackexchange datasets. We acknowledge that while other websites may vary slightly in behaviour. However, most of the ideas presented here will apply to all forum based discussion websites. As a side effect of choosing this dataset, we never had a reference pre- summarized dataset, such as one available with the CNN/Daily Mail dataset. While this initially acted as a hindrance, not allowing us to design pre-trained models for extractive summarization, the sheer volume and variety of theforum websites would have meant we would need to essentially create an entirely new extractive model for every single website, which could wind up being an absurd amountof memory required for storage, as opposed to representing each website using only four float values (a,b,c and W2a).

Lastly, results may be improved further by switching out the basic extractive ranking and topic modelling algorithms for more advanced ones. The purpose of this project was to observe if including user scores and the naturally available hierarchy data present in forums improves extractive summarization results. Since that has been confirmed, this can now be layered upon any number of other advanced extractive summarization techniques to improve results by including that necessary human input.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have shown that taking valuable human input into consideration in the form of user scores can positively impact the performance of extractive summarizers,leading to a more representative output when compared to basic TextRank. We used the naturally occurring structure ofthe thread to add context and thus improve comprehensibility. Finally, we performed some basic topic modelling to allow the user to remove informational noise if they choose to do so. When compared to baseline extractive summarization techniques via TextRank, our methods produced more representative, comprehensive and cohesive results according to our tests.

Since the basic procedure is now understood, it is simple enough to improve upon it where possible.

More advanced untrained extractive summarization solutions can be used in place of basic TextRank. In case the target domain is specificenough, a custom pre-summarized dataset can be utilized to bring about more focused pre-trained extractive summarizing model(s). A larger base for testing will allow for betterjudgement of user score weightage through W2a. Testing to find the best granularity for the values in W2b(x) across websites will also be possible with a larger testing base. We worked with the stackexchange datasets. Working with datasets that provide even more data, such as post views, or forums with more complicated hierarchies, forums that include replies and other features, can lead to even better performance when the additional data is considered.

REFERENCES

[1]. A Survey on Extractive Text Summarization, N.Moratanch,S.Chitrakala, IEEE International Conference on Computer,
[2]. Communication, and Signal Processing, 2017
[3]. Topic-Focused Summarization of Chat Conversations, 35th EuropeanConference on Information Retrieval, 2013
[4]. TextRank: Bringing Order into Texts, Rada Mihalcea and Paul Tarau, Conference on Empirical Methods in Natural Language Processing, 2004