Extractive Review Summarization Framework for Extracted Features

Palak Bansal, Somya, Nazar Kamaal, Shreya Govil, Tameem Ahmad

Abstract*—In the information age, the growth of e-commerce has brought the products' sale and purchase online and many of the customers prefer to buy it online. To support this preference the users' reviews of the products plays an important role. So, online merchants wish to take the reviews; experiences of the user, to enhance their business and revenue. Popular and trending products may attract large number of reviews. Further, many of which could be elongated. Extracting useful information with efficiency and accuracy from these so many reviews, of which there are some very long, is a challenging task. This work is an attempt to summarize the customer reviews on products into more useful and shorter version that can help another users' decision. Reviews available online are crawled for product, each time after extraction, first identification of features of the product will be done and hence polarity will be detected i.e. either a review is positive review or a negative review. After the calculations, summarization of all the features of the product will be generated.

***Index Terms*—Text summarization, text mining, opinion mining, extractive summary, abstractive summary, feature identification**

1. INTRODUCTION

Automatic text summarization is the process to present the overall summary of all the documents or the text so that it can represent the gist of the all. It may reduce the original text into a smaller edition, which contains most important aspects and favors the user to quickly grasp the large amount of information in a short time.[11] It uses the machine learning and data mining capabilities and involves interpretation, transformation and generation.

Generally, extraction and abstraction are the two approaches to automatic text summarization. Selecting part or whole word, phrases or sentences from the existing source(s) and combining them to form the summary is the extraction based summarization. On the other hand, abstraction based approach makes an internal semantic representation that involves paraphrasing and shortening parts of the source(s) and hence it is summarized using natural language generation techniques which might exactly look like the human generated summary.

According to number of source documents processed, summarization can be classified into single-document and multi-document summarization. Documents can also be classified on basis of external resources as knowledge-rich or knowledge-poor. Knowledge rich summarizers uses sources like Wikipedia, Word Net etc, For query oriented summarization, summary is generated according to information related to query. In update summarization, the summarizer uses recent trends for summary generation.

Today's world is all about information, most of which is available online.[12]Most information in the world is stored in text because of its permanence an ability to be shared. The World Wide Web contains trillions of documents containing information and is growing at an exponential rate day by day. For example, according to Google, there are 130 million books in the entire world. Most of the information is not accessible easily because it's impossible for humans to read each and every book or research papers in the world or the thousands of reviews about an item on a website. This is the reason for the books to have summaries at the end, research papers having abstracts, and the existence of Wikipedia . However, in order to make a summary, a person had to manually compile information and write it, which is a time-consuming task. We believe that building a system that can automatically build summaries for us would allow us to access information in a more digestible format and save a lot of time for humans . Recent years have seen the development of numerous summarization applications.[2][14][7]Achieving this is challenging for many reasons for the reasons of uncertainty of knowing exactly what the user's information needs are.[15] These factors have led to the development of automatic summarization systems. And since online items and their reviews are the main issues these days, building an extractive based system can help the customers to take decisions efficiently about a particular product.[3]

Also, summarization is a typical hard problem in the domain of Natural Language Processing as one has to fully understand the text to do it.[10] that requires semantic analysis, processing, and deduced interpretation. Incorporating multiple sources of knowledge and applying frame semantics at the noun phrase, sentence, and document level has been focused by most of the recent works in Natural Language Processing. So, proposing an efficient system which can somehow reduce the problem of large text reading by summarizing it feature wise will be our objective in this project and it can help the customers to take decisions effectively and saving their time will lead to some scope in improvement.[5][8]

Revised Version Manuscript Received on May 29, 2019.

PalakBansal, Department of Computer Engineering, Z. H. College of Engineering & Technology, A.M.U., Aligarh, India

Somya, Department of Computer Engineering, Z. H. College of Engineering & Technology, A.M.U., Aligarh, India

NazarKamaal, Department of Computer Engineering, Z. H. College of Engineering & Technology, A.M.U., Aligarh, India

ShreyaGovil,Department of Computer Engineering, Z. H. College of Engineering & Technology, A.M.U., Aligarh, India

Tameem Ahmad, Department of Computer Engineering, Z. H. College of Engineering & Technology, A.M.U., Aligarh, India

This paper is organized in five sections. It begins with section 1 which gives a brief introduction of the work along with a brief idea about the approaches to the summarization work. In section 2 we have discussed some of the literatures about automatic text summarization. Next chapter i.e. section 3 is dedicated to all the processing and implementation that has been performed so far to achieve desired output. Sectionr 4 discussed the experimental results and section 5 is conclusion with its future scope. References are cited in the last section.

2. LITERATURE REVIEW

Automatic text summarization is a field of interest for many researchers and a lot of research work has been done in this field in recent years. Many automatic text summarization systems have been developed which have successfully reduced the size of text without the loss of the information.

The paper presented by Eduard Hovy and Chin-Yew-Lin[6] described the various characteristics of source text as well as the characteristics of the summary produced as output. In this paper, they first described the architecture of SUMMARIST and provided details on the evaluated results of several of its modules. They also described some preliminary experiments in this regard.

George, Michael. "APPROACH FOR THICKENING SENTENCE SCORE FOR AUTOMATIC TEXT SUMMARIZATION"[4] proposed a system that worked on sentence extraction based text summarization task using the graph of 5 main tasks: Text pre-processing, term N.R weight determination, term relationship exploration, sentence ranking and based algorithm to calculate important sentences in document and most important sentences were extracted to generate document summary. They first pre-processed the data, then they built graph which represented a sentence as a node with all its properties, finally they used the sentence ranking algorithm to produce the summarized text.

N.P Vadivukkarasi, Dr. B.Jayanthi described in their paper 'Product review Ranking Summarization' N.Q(2015) [1] where they proposed a system consisting summary generation. They used the aspect based opinion mining in the reviews given and presented feature wise summarised results. Processes involved were: Sentences from large review dataset were extracted and review words except noun and verbs were identified and then sentiment classification was used to classify them under negative and positive. Finally, weight was calculated for each review word and provides product ranking. They used Decision Tree pattern Extraction Algorithm (DTPE) that identifies the repeated words to omit them.

3. PROPOSED METHOD

A. *FRAMEWORK*

The framework which follows the approach given by us in order to carry out the summarization is shown in fig1.

INPUT: The inputs to our summarizer system are the reviews crawled from a ecommerce website.

OUTPUT: The output is the polarity based summary of all the reviews .The summarization is done in four main steps:

1) Preprocessing,
2) Feature Extraction,
3) Clustering, and
4) Opinion Extraction.

These steps are further divided into multiple sub-steps.

The system crawls and downloads all the reviews of a particular product using scrapper. These reviews are then stored in a plain text file.

This text file is then parsed and preprocessing is done which includes conversion of the crawled reviews into lower case. After which, they are separated as tokens and accordingly the sentences are broken. The next step is the Feature extraction which includes POS tagging of the above extracted tokens. Various features are extracted using grammar and then these are combined with manually extracted features. The redundancy of features is removed. Then the reviews are extracted and classified according to the features and clusters are formed. The system then finds the infrequent features and filter it.

Sentiment analysis of all the reviews is carried out, which give them a polarity score. For each feature, the statistics of positive, negative and neutral reviews is displayed and overall polarity of each feature is calculated according to the statistics. The most positive, most negative and a neutral sentence (if present) constitute the summary of the particular feature.

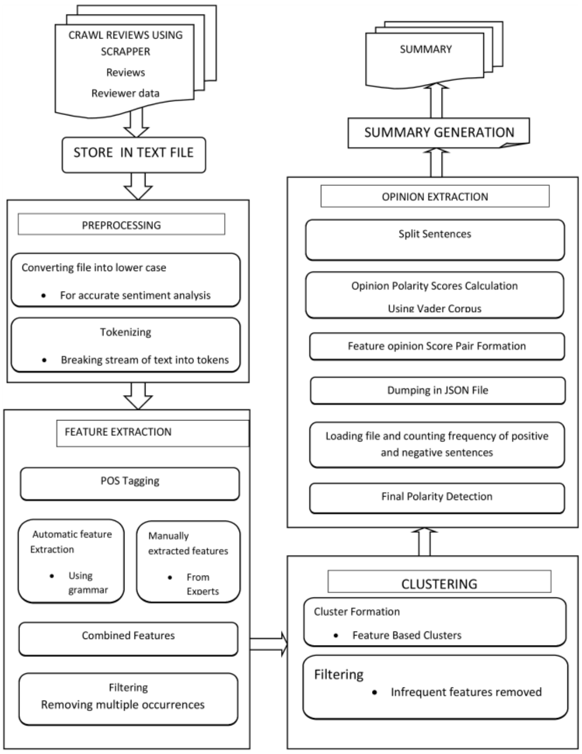

CRAWL REVIEWS USING SCRAPPER
Reviews
Reviewer data
STORE IN TEXT FILE
PREPROCESSING
Converting file into lower case
For accurate sentiment analysis
Tokenizing
Breaking stream of text into tokens
FEATURE EXTRACTION
POS Tagging
Automatic feature Extraction
Using grammar
Manually extracted features
From Experts
Combined Features
Filtering
Removing multiple occurrences
CLUSTERING
Cluster Formation
Feature Based Clusters
Filtering
Infrequent features removed
OPINION EXTRACTION
Split Sentences
Opinion Polarity Scores Calculation
Using Vader Corpus
Feature opinion Score Pair Formation
Dumping in JSON File
Loading file and counting frequency of positive and negative sentences
Final Polarity Detection
SUMMARY GENERATION
SUMMARY

Figure 1 Proposed model

B. *PSEUDOCODE*

Class start:

- Open the crawled reviews file in read mode.
- Read the file and convert it into the lower case.

```
Class preprocessing:
    •  tokenizing
    •  Using the tokenizer, convert these sentences into
       tokens.
    •  Tagging:
         ▪ Using POS tagger, mark every extracted
           token corresponding to its part of speech.
         ▪ Return tags
    •  Extraction:
         ▪ Add the manually extracted features from
           experts into the feature list.
         ▪ For all tags:
           Use the grammar and extract the features.
         ▪ For all the extracted features:
           Retain only those features that appear more than
    a minimum threshold.
         ▪ Remove duplicate features by using set.
         ▪ For all extracted features:
    Use regular expression to extract sentences that contain a
particular extracted feature add the word and its respective
sentences into list
      •  for all sentences in the list
    Remove the features who have sentence count less than a
minimum threshold
Class sentiment:
    •  for all the features in list
         ▪ For all the sentences of a given feature:
         ▪ Split each sentence into simpler sentences using
           grammar
         ▪ Calculate the polarity score for each sentence
           using sentiment analysis function of vader
           corpus.
    •  Dump the data, i.e. the feature, sentence and its
       senti-score into json format.
    •  for each feature in the json file
         ▪ Calculate the number of negative, positive and
           neutral sentences.
         ▪ Determine the overall polarity of the feature.
Class summarization:
    •  load the json file.
    •  For each word in the json file
    present as summary, the most negative, the most positive
and one neutral sentence from the review sentences.
```

C. CRAWLING THE REVIEWS

The first step is to crawl the reviews from a particular website of a particular product using a scrapper and storing those reviews in a text file. Here, we have crawled the reviews of Snapdeal from https://www.snapdeal.com/ as shown in fig2 and have taken mobile phones as a product for testing purpose using BeautifulSoup and Urllib2.

For testing, we have crawled the reviews of various phones such as Iphone, Lecco LE2, Motorola Moto G.

D. PREPROCESSING

This is the second step of the proposed automatic review summarization system. The step is to mine various features of a product that have been discussed on in the reviews by the customers by loading it from the text file. Data mining and Natural Language pre-processing techniques are used for this task.

Stemming is considered an important part of preprocessing data. But, in our case the results we obtained after stemming were not efficient. Various words were inaccurately stemmed.

Apart from stemming another major step included in preprocessing is the removal of stop words. Since the output of our system has to be a summary of the reviews. Removal of stop words was again an unnecessary overhead.

E. TOKENIZING

A tokenizing is the task to split the given text or string into substrings (pieces) on the specified string. Each chopped piece is referred as token. The process is shown in the following code:

```
>>> import nltk
>>>sentence = “The camera is really good”
>>>tokens = nltk.word_tokenize(sentence)
>>>tokens
```

F. FEATURE EXTRACTION

1. *PART-OF-SPEECH TAGGING (POS)*

PoS tagging helps us in extracting the features of a product. They are by and large nouns or noun phrases in text of the reviews. We used the NLTK POS tagger to parse the text of each of the review to break them text into sentences and to generate the POS tag for each word. Further these tagged sentences are saved in the along with the POS tag information of each word in the sentence. To improve, fuzzy logic can be used to deal with the problem of word variants and misspellings.

```
>>>import ntk
>>>from nltk import pos_tag
>>>from nltk import word_tokenize
>>>sentence”The camera is really good”
>>>tags = pos_tag(word_tokenize(sentence))
>>>tags
```

2. *PARSE TREE*

Parse tree is the diagrammatical representation of grammatical structure of a sentence i.e. this tree structure gives the idea about the syntax of a sentence.

```
>>>import nltk
>>>sentence=“The camera is really good”
>>>from nltk import pos_tag, word_tokenize
>>>tags = pos_tag(word_tokenize(sentence))
>>>pattern = “NP: {<DT>?<JJ>*<NN>}”
>>>NPChunker = nltk.RegexpParser(pattern)
>>>result = NPChunker.parse(tags)
>>>result.draw()
```

3. *FEATURES EXTRACTED*

Here, our focus is to identify the features of the product specially the frequent features that are referred many times by many users. It is done by using regular grammar and other manually extracted features are also added. These manually added features are brought by the experts.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

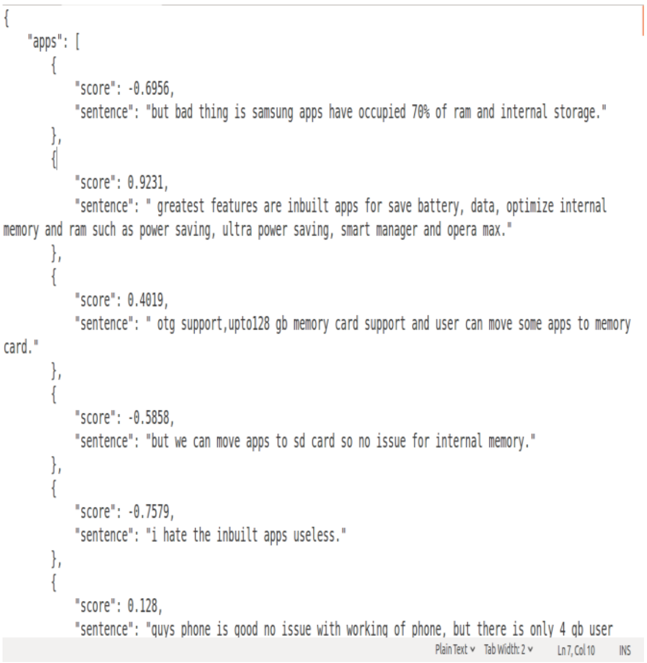
The extracted nouns and proper nouns in our case are considered as features in addition to the manually extracted features. These particular features will be filtered in case of duplicacy and will be further passed on to make clusters with sentences and scores.

G. CLUSTERING

Clustering is the process of organizing objects into groups whose members are similar in some way. This step involves extracting sentences on the basis of extracted features and forming clusters as feature-sentence pair. Various infrequent features are also extracted which are removed using threshold limit.

H. OPINION EXTRACTION

The next important step is opinion extraction of each review i.e. identification of orientation or the tone of a reviews, i.e., positive, negative or neutral. Opinion extraction or Mining is also known as Sentiment analysis.



```
{
    "apps": [
        {
            "score": -0.6956,
            "sentence": "but bad thing is samsung apps have occupied 70% of ram and internal storage."
        },
        {
            "score": 0.9231,
            "sentence": " greatest features are inbuilt apps for save battery, data, optimize internal
memory and ram such as power saving, ultra power saving, smart manager and opera max."
        },
        {
            "score": 0.4019,
            "sentence": " otg support,upto128 gb memory card support and user can move some apps to memory
card."
        },
        {
            "score": -0.5858,
            "sentence": "but we can move apps to sd card so no issue for internal memory."
        },
        {
            "score": -0.7579,
            "sentence": "i hate the inbuilt apps useless."
        },
        {
            "score": 0.128,
            "sentence": "guys phone is good no issue with working of phone, but there is only 4 gb user
```

Figure 2 Performing sentiment analysis

In fig 2, we see how sentiment analysis is done on the preprocessed list and the compound polarity score of all extracted reviews is calculated. The extracted reviews and their senti-score are then dumped into a json file. After loading the file, the frequency of positive and negative sentences is counted and hence the overall polarity of the feature is determined.

I. SUMMARIZATION

The goal of any automatic text summarization is to reduce a text document to a summary that still contains key points of the original text.

After all the previous steps, the final feature-based review summary can be produced. For each identified feature, related opinion sentences are then put into positive, negative and neutral categories according to the opinion sentence's orientations. The most negative or positive sentence of the cluster is picked up and presented as summary. For neutral, one sentence at random is chosen and presented. We initially considered taking the average score of each cluster and then presenting the review with score nearest to the average score as overall polarity of the feature, but eventually averaging out the sentiment score and presenting a summary did not proved effective. So we came up with another solution which gave us efficient summaries. We counted the frequency of senti-scores of each cluster and selected the most negative and most positive review Hence, this generated summary proves to be better way for the customers to reduce their time in reading all the reviews online and helps them to take their decisions accurately and efficiently. Some underlying intermediate steps are also used as discussed below:

1) **Stemming:** It is a method of reducing words to root form. It is not compulsory for a stem word to be identical to the root of the word. Many search engines treat words with the same stem as synonyms. For example: Stemming reduces the words "fishing", "fished", and "fisher" to the root word, "fish". But in addition to the advantage in using it, we faced some difficulties as well. In many cases, the root word formed was not always the desired one. For example: It also reduced the words like “Important” to the root word “import” which was totally incorrect.

2) **Stop word removal:** These are those most common words in any language which are filtered out before we do processing of NLTK data. There is no single list of such words but we can chose any group of words for this purpose. For example: words like the, is, at, which, on etc. But there is problem which we faced during splitting of sentences particularly in names like “Take that”, ”The Who”, etc.

3) **Splitting of sentences:** We have implemented sentence breaking to increase the efficiency of sentiment analysis and getting more accuracy but we various problems were faced during the writing of its grammar.

4) **Word-sense disambiguation:** It is an open problem which helps in identifying the real sense of a word as single word can have multiple meanings in a sentence. This requires a dictionary as an input to specify the senses. Various steps we faced problems due to this issue. Sentiments if various words were not extracted correctly.

4. EXPERIMENTAL EVALUATION

According to our proposed framework of feature based summarization, we now evaluate this framework from three perspectives:

- Accuracy of summary produced.
- Precision
- Recall

We have conducted our experiments using four types of mobile phones.

- Moto-g
- Samsung galaxy
- Apple Iphone
- Leeco Le2

The reviews have been crawled from Snapedeal.com for testing purpose.

The way we produced the summary according to our framework i.e. feature and opinion based all together has never been implemented and there is no such tool to evaluate it. So comparing with summaries of other frameworks was not possible.

So, here we calculated the efficiency of our own system by defining the following metrics:

- **True Positives (TP)**: number of positive reviews in summary, labelled as such.
- **False Positives (FP)**: number of negative reviews in summary, labelled as positive.
- **True Negatives (TN)**: number of negative reviews in summary, labelled as such.
- **False Negatives (FN)**: number of positive reviews in summary, labelled as negative.

We can define accuracy, precision and recall as follows:

1. Accuracy=(TP+TN)/(TP+TN+FP+FN)

2. Precision=TP/(TP+FP)

3. Recall=TP/(TP+FN)

	Moto-G	*Samsung galaxy*	*Apple Iphone*	*Leeco Le2*
Accuracy	**81.25%**	**88.88%**	**90.90%**	**85.71%**
Precision	**0.833**	**1**	**1**	**1**
Recall	**0.909**	**0.8571**	**0.833**	**0.8**

Table 1 Evaluation of results

5. CONCLUSIONS

With the growing availability of review sites for products available online, selecting the best possible product from verity of products has become cumbersome process for the user. If a user goes through only a few top reviews because he has no time to read all the reviews, then he may get a biased and incomplete view about the product. So this work is an attempt to develop a system for feature based summarization of customer's reviews for products available online.

Every individual has its own perspective of viewing things. The reviews that we crawled had a lot of diversity.[13][15]Extracting features was itself a tough task, as our approach was to define a grammar which failed in various cases of typing errors.After the features were extracted, sentiment analysis was carried out and clusters were made according to the polarity scores. Lastly, the summary that we presented provided the user a feature wise unbiased extraction based summary of the reviews. According to the experimental evaluation, we can observe that our proposed framework was quite promising and proved to be a better system for feature based summarization. A perfect system is when precision= recall=1. Here, our result says it all. We have reached quite a good summarization system.

6. FUTUREWORK

Sentiment Analysis and summarization are huge fields and research is still going on and have become an important research area of NLP as it is a really big task for a machine to learn and understand the natural language.[7][8] Now-a-days many systems focusing on summarization have been implemented but efficiency has always been an issue. Further changes in our proposed framework can help improve efficiency to a different level in order to build a better system[9][16][17]. Here, we have tested this system on a particular product. Extension to different products at a time can be taken into account. Furthermore, the grammar we have used for feature extraction can be modified and can be converted into a semi-supervised technique to make the system learn on its own and extract the relevant features. In future works, instead of just extraction, abstraction based summarization can also be implemented.

REFERENCES

1. P Vadivukkarasi, Dr. B. Jayanthi "Product Review Ranking Summarization" International Journal of Advanced Research in Computer Engineering and Technology(IJARCET) Volume 4 Issue 7 (2015)
2. Vijay B. Raut, Prof. D.D. Londhe "Survey On opinion Mining and Summarization of User Reviews on Web" International Journal Of Computer Science and Information Technologies Vol 5(2),2014
3. Sheeba, J. I., and K. Vivekanandan. "Improved sentiment classification from meeting transcripts." IJCSI International Journal of Computer Science Issues 9.5 (2012): 1694-0814.
4. George, Michael. "APPROACH FOR THICKENING SENTENCE SCORE FOR AUTOMATIC TEXT SUMMARIZATION."
5. Hu, Minqing, and Bing Liu. "Mining and summarizing customer reviews." Proceedings of the tenth ACM SIGKDD International conference on Knowledge discovery and data mining. ACM, 2004.
6. Hovy, Eduard, and Chin-Yew Lin. "Automated text summarization and the SUMMARIST system." Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998. Association for Computational Linguistics, 1998.
7. Tadano, Ryosuke, Kazutaka Shimada, and Tsutomu Endo. "Multi-aspects Review Summarization Based on Identification of Important Opinions and their Similarity." PACLIC. 2010.
8. Seki, Tsunehito, Kazutaka Shimada, and Tsutomu Endo."Acquisition of synonyms using realtions in product." Systems and Computers in Japan(2007), 38(12), 25-36
9. Zhang, Wei, and Jianyong Wang. "Integrating Topic and Latent Factors for Scalable Personalized Review-based Rating Prediction." IEEE Transactions on Knowledge and Data Engineering 28.11 (2016): 3013-3027.
10. Naing, May Thu, and Aye Thida. "Text Summarization System Using Myanmar Verb Frame Resources." (2018).
11. Suanmali, Ladda, NaomieSalim, and Mohammed Salem Binwahlan. "Feature-based sentence extraction using fuzzy inference rules." 2009 International Conference on Signal Processing Systems. IEEE, 2009.
12. Ahmad, Tameem, Shamim Ahmad, and Mohammed Jamshed. "A knowledge based Indian agriculture: With cloud ERP arrangement." 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, 2015.
13. Varshney, Vanshika, et al. "Recognising personality traits using social media." 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). IEEE, 2017.
14. Cui, Weiwei, et al. "Textflow: Towards better understanding of evolving topics in text." IEEE transactions on visualization and computer graphics 17.12 (2011): 2412-2421.

15. Zhan, Jiaming, Han Tong Loh, and Ying Liu. "Gather customer concerns from online product reviews–A text summarization approach." Expert Systems with Applications 36.2 (2009): 2107-2115.
16. Mann, William C., and Sandra A. Thompson. "Rhetorical structure theory: Toward a functional theory of text organization." Text-Interdisciplinary Journal for the Study of Discourse 8.3 (1988): 243-281.
17. Gong, Yihong, and Xin Liu. "Generic text summarization using relevance measure and latent semantic analysis." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication