Automatic Summarization on Aggregated Search Results

Rohit Arvind Chakrapani Rakesh Balabantaray Pallavi Verulkar

Computer Science and Engineering
International Institute of Information Technology (IIIT),
Bhubaneswar, Odisha, India.

ABSTRACT
Aggregated search is the task of integrating results from potentially multiple specialized search services, or verticals (images, videos, news, weather-forecasts, etc), into the web search results. However these results are mainly in the form of hyperlinks. As a result, the user has to go through each link to find and organize its relevant and focused data. In this paper, we are proposing a tooltip-like feature to the user which will provide the summarized content of the page he/she wants to surf i.e. whenever the user hover its mouse over the response link; the tooltip will appear, giving the relevant summary about the current page. Thus, by going through the summary, the surfer can decide whether the link is relevant to navigate or jump to some other links. As a result, he/she can save his/her time rather than clicking on each link, going through the content of corresponding page and then deciding whether the page is relevant to the queried response or not. Here, a simple methodology is proposed for providing the summary dynamically regarding the content of the web page which is based on an Automatic Summarization using Key-phrase extraction method.

General Terms
Information Retrieval, Data Mining, Knowledge extraction.

Keywords
Aggregated search, Inverted index, Term-document matrix, summarization, Precision scores.

1. INTRODUCTION
A vital role of a web search engine is to display links to relevant web pages for each issued user query. Data dispersal, lack of focus and ambiguity are just some of the limitations of traditional ranked retrieval system. In recent years, search engines have extended their services to include search, so-called vertical search, on specialized collections of documents, so-called verticals, focused on specific domains (e.g., news, travel, shopping) or media/genre types (e.g., image, video, blog). In general, the user looking for relevant material browses and examines the returned documents to find those that are likely to fulfill his need. If lucky, the user will find in this list the document that satisfies completely his/her need. However, when one document alone is not enough (i.e., the relevant information is scattered in different documents), the user has to collect and aggregate information coming from different documents to build the most appropriate response to his/her need. Combining this different information to achieve, at the same time, better focus and better organization within the search results is the scope of aggregated search, which is defined to go beyond the uniform ranking of snippets. Thus aggregated search addresses the task of searching and assembling information from a variety of information sources on the web (i.e., the verticals) and placing it in a single interface. In this paper, a simple issue regarding aggregated search is discussed. We have proposed a method to provide dynamically the summarized content on each of the links for the aggregated search results that has been retrieved. As a result, the user will go through that summarized content and find whether the result is relevant or not.

2. RELATED WORKS
All Aggregated search as a research area, has derives its importance from many previous works such as federated search ,meta-search, question answering, semantic search, and entity-oriented search. Aggregated search seeks relevant content across heterogeneous information sources, the verticals. Searching diverse information sources is not new. Federated search (also referred to as distributed information retrieval) [1,2] and meta-search are techniques that aim to search and provide results from various sources .In federated search, a user submits a query, and then may select a number of sources, referred to as resources, to search [3]. These resources are often standalone systems (e.g., corporate intranets, fee-based databases, library catalogs, internet resources, user-specific digital storage).Examples of federated search systems include Funnelback, Westlaw, FedStats. Bringing federated search to the Web led to two different paradigms, meta-search and aggregated search. A meta-search engine is a search engine that queries several different search engines, and combine results from them or display them separately. However, in aggregated search, the information sources are powered by dedicated vertical search engines, all mostly within the remit of the general web search engine, and not several and independent search engines, as is the case with meta-search. In addition, the individual information sources in aggregated search retrieve from very different collections of documents (e.g. images, videos, news) which are called verticals and then all vertical results are aggregated into web search result [4,5].

2.1 General Framework of Aggregated Search
In a broad interpretation, aggregated search concerns at the same time the retrieval and the assembly of search results. Thus I have gone through a proposed unified framework for aggregated search that facilitates the analysis of the many and diverse approaches related to aggregated search. The framework [6], shown in Figure 1, involves three main components, namely, Query Dispatching (QD), Nuggets Retrieval (NR), and Result Aggregation (RA).

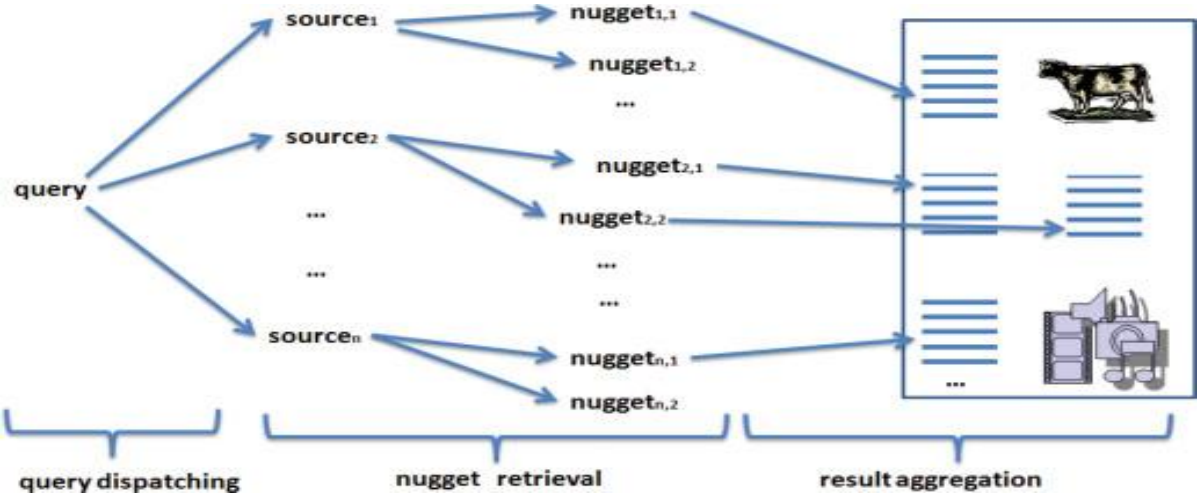

Figure 1: General Framework of Aggregated Search [6]

2.2 Cross-verticals Aggregated Search

Cross-vertical aggregated search attempts to achieve diversity by presenting search results from different information sources, so-called verticals (image, video, blog, news,etc.), in addition to the standard web results, on one result page [4]. In cross-vertical aggregated search it is easy to identify the components of our general framework for aggregated search. Query dispatching will correspond to the selection of sources. Each source will perform its nugget retrieval process and finally retrieved nuggets (e.g., Web pages, images, videos) will have to be assembled in one interface. Although the tasks are well distributed, the problem is far from being solved. It is not easy to decide which sources should be used and how the retrieved results should be assembled and presented.

3. AUTOMATIC SUMMARIZATION

Automatic summarization is the task of taking an information source, extracting the content from it and presenting the most important content to the user in a condensed form and in a manner sensitive to the user's or application's need. As the problem of information overload has grown, and as the quantity of data has **increased**, so has interest in automatic summarization. This process is mainly exploited for the tooltip features showing the summarized content, which will get displayed while the user hover its cursor over the link of the retrieved aggregate search results. This area is highly interdisciplinary and related with natural language processing, artificial intelligence, information retrieval [7], information extraction, statistics and cognitive psychology. Generally there are two approaches to explore automatic summarization, namely Extraction and Abstraction based methods.

3.1 Extraction based summarization

Two particular types of summarization often addressed in the literature are keyphrase extraction, where the goal is to select individual words or phrases to "tag" a document, and document summarization, where the goal is to select whole sentences to create a short paragraph summary. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary [9,10,11].

3.2 Abstraction based summarization

Abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Research into abstractive methods is an increasingly important and active research area, however due to complexity constraints; research to date has focused primarily on extractive methods. In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require the use of natural language generation technology, which itself is a growing field.

4. PROPOSED WORK

The idea of aggregated search was explicitly introduced as universal search in 2007 by Google. However they implemented it in 2011.Now almost all major search engines are providing with the aggregated search results for any query. Furthermore, they also provide some text content below the links that actually contain the topmost few lines of the document which is mostly non-relevent to the user to determine its relevancy. So my idea is to modify this concept by providing some relevant summary about the document contained within the link. Henceforth, whenever the user hover its cursor over the link, a tooltip like feature will appear providing the summarized content about the page for the corresponding link. If the user finds it relevant, then he/she may navigate to that page by clicking the link; otherwise jump to other links. This leads to save the surfing time by avoiding the click operations on the irrelevant link, waiting for that link document to open, and then going through the document to find it is relevant or not. Thus in this paper, we have proposed an algorithmic work for finding the automatic summarization of the web document for the retrieved links of the aggregated search results dynamically displayed on the interface which is discussed in below subsections. Here, the query dispatching and nugget retrieval is performed with the help of Solr search engine enterprise where the large number of documents is crawled and indexed using the integration of Apache Nutch and Solr.

4.1 Algorithmic work

Below is the proposed algorithmic work for finding the relevant summarized content of the particular web document which is a part of dynamically retrieved aggregate search results displayed on the response interface. Algorithm used for above summarization includes following steps:

1. Document Reconstruction: Transform text files of HTML and XML or JSON format into plain text.
2. Transformation of the document to the separate sentences.
3. Removal of Stop words and special characters.
4. Parsing and extracting unique words with its corresponding frequency and significant weight.
5. Extracting the sentences according to its weight factor and ranking [11].
6. Display Summary as an output.

Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called stop words [7]. Some of them are listed below.

a an and are as at be by for from

has he in is it its of on that the

The special characters such as &, %, !, #, @, ?, etc. are also removed from the sentences along with the above stopwords. Thus the sentences now only contain the significant words in it, which is to be processed further.

```
For each Document doc:
{
  Read doc:
  Call convert_doc_to_sentence( ) method.
  For each sentence ϵ doc
   {
     Call remove_stopwords( );
     Form the arraylist of unique words;
     For each word:
     {
       Calculate the term-frequency(tf) and corresponding
        weight by calling the set_weight(token) method.
      }
    Retrieve the sentence having token of maximum
      weight.
   }
  Call display_summary( );
}
void display_summary( )
{
Retrieve the few most significant sentences w.r.t. its
ranking score and display it.
}
```

4.2 Results and Evaluation

4.2.1 Results

The result part consist of the snapshots about the user firing the query in web interface, getting the response for the fired query, same query is being processed by the Solr in the backend and finally the summarized output for the retrieved links of aggregated search results is displayed. These results are shown from figure 2-5.

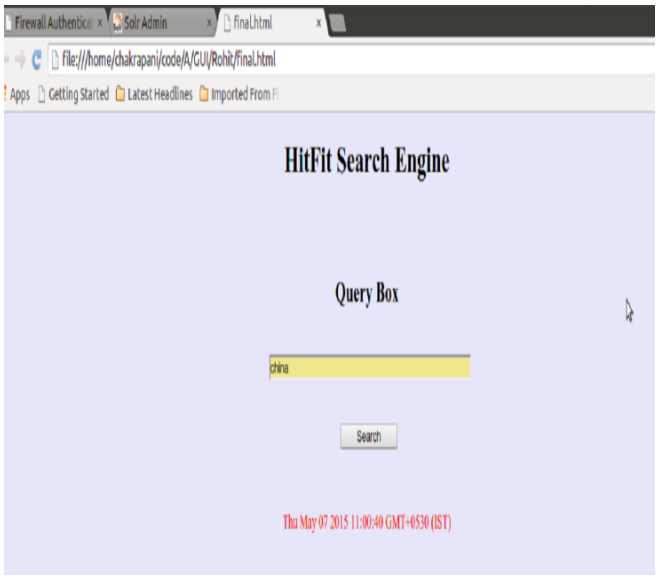

Figure 2: Interface for firing the user query

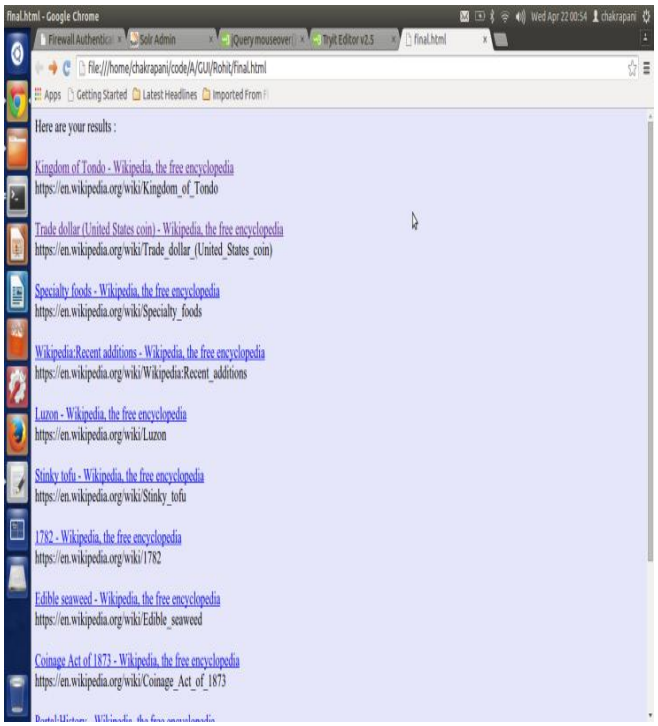

Figure 3: Getting the Response for the query

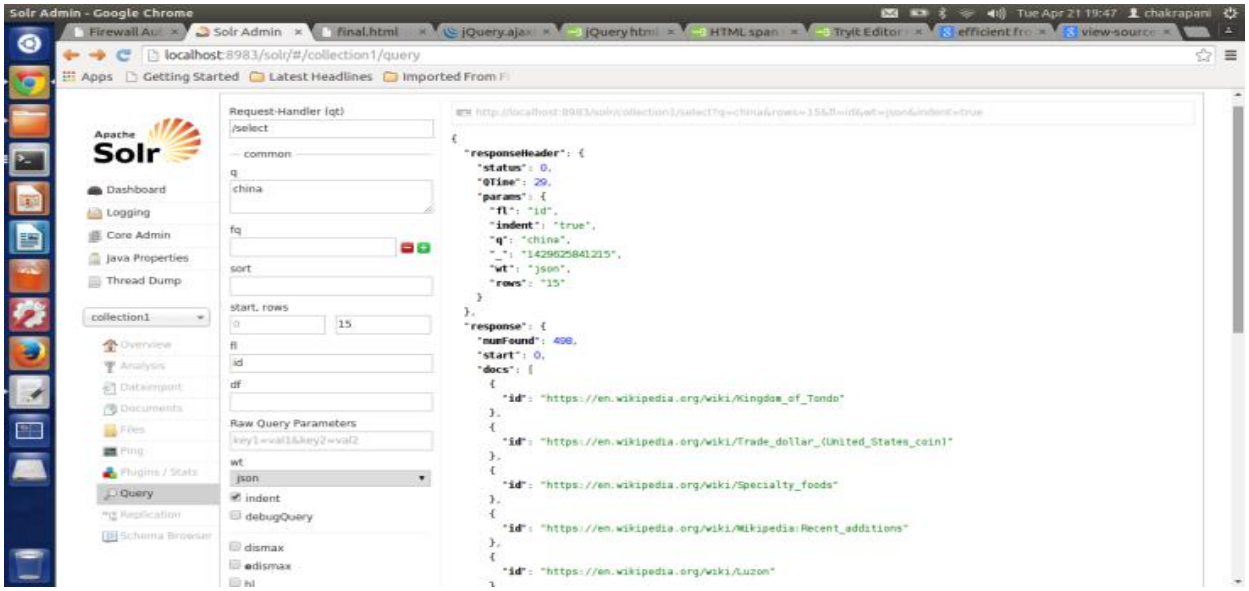

Figure 4: Solr backend process of searching

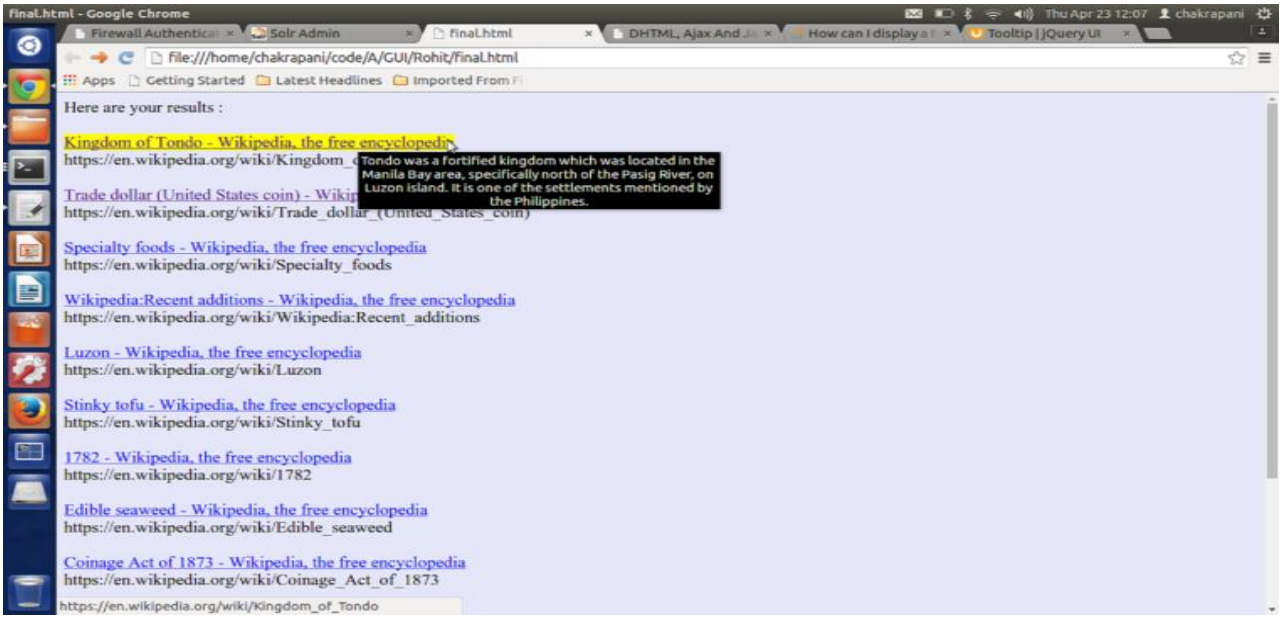

Figure 5: Summarized output for the searched results

4.2.2 Evaluations

Assumption: Evaluation of the retrieved results is divided into four equivalence classes with some assumed score value is assigned in the range of [0,1] for the particular query. Here I took help of some of my friends and given the job of a assigning the relevance score to the retrieved link as the response for their particular desired query. Thus they acted as evaluator and assigned the relevance score as given below in the Table No.1.However, there are other methodologies discussed for evaluating aggregated search results [8].

Assumed values for the above four equivalence classes are given below.

Relevant = 1

Partial relevant = 0.5

Indirect Link Relevancy = 0.25

Non relevant = 0

Table 1. Relevance score of the retrieved link for the particular query term.

Query Term	*Link 1*	*Link 2*	*Link 3*	*Link 4*	*Link 5*	*Link 6*	*Link 7*	*Link 8*	*Link 9*	*Link 10*
China	0.5	0.25	0.5	0.25	0.25	0.25	0.25	0.25	0	0
Nutch	0	0.25	0.25	0.25	0.25	0.25	0.25	0.5	1	0.5
Narendra Modi	0.25	0.25	0.5	0.25	0.25	0	0	0	0	0.25
Nepal earthquake	1	0.5	0.25	0.5	0.25	0.25	0.25	0.25	0	0.25
IPL	0.5	0.5	1	0.5	0	0.25	0	0.25	0	0
Data mining	0	0	0.5	0.25	0.5	0	0	0	0	0.5
Argentina	0.5	0	0.25	0.5	0	0	0	0	0	0.5
XML	0.25	0.5	0.5	0.25	0	0	0	0.25	0	0
Salman Khan	0.5	0.5	0.5	0.25	0	0	0	0	0.25	0
Mumbai Tourist spot	0.25	0.25	0	0	0	0	0	0	0	0

Precision is given by,

Precision(P) = #(relevant items retrieved) / #(retrieved items)

But we have formulated new calculation for evaluating information retrieval system i.e. precision score for a particular query.

Precision score = Sum of relevance score of each link / #(retrieved link)

$$Ps@(n) = (\sum_{i=1}^{n} Li)/n \quad \text{where;}$$

Li = relevance score of each link.

Table 2. Precision Score P@(5) for the Query Term

Query Term	*Precision Score*
China	0.25
Nutch	0.425
Narendra Modi	0.175
Nepal earthquake	0.35
IPL	0.3

Here only the precision is calculated due to limited data set being crawled i.e. around 8000 documents and also we are not accompanied with already relevant and non-relevant documents about the particular query term; so there is no evaluation concerning about recall factor. Due to limited data set in the collections, we are getting lower precision values as per issued query by the user. The evaluator entered their query term in the interface and according to their information need have allocated some assumed score values to the retrieved links of aggregated search results from which precision score is calculated. Table No 2 provides the precision score for the first five query term with regard to the Table No 1.Thus by increasing the number of documents in the collection, the precision score can easily be increased for the particular query term of a particular domain i.e. verticals.

5. CONCLUSION

Aggregated search aims to facilitate the access to the increasingly diverse content available from the verticals. It does so by searching and assembling relevant documents from a variety of verticals and placing them into a single result page, together with standard web search results. The goal is to best layout the result page with the most relevant information from the conventional web and verticals. Most current search engines perform some level of aggregated search. We expect this to be a rich and fertile research area for many years to come. In aggregated search, the final result is not necessarily a ranked list of documents or snippets, but it can be whatever combination of content that can turn useful to the user. Aggregated search can thus propose more focus, more organization, and more diversity in search results. Automatic Summarization on aggregated search results helps the user to surf the focus and relevant content without wasting their time to go through each links.

6. FUTURE WORK

The most challenging issues concern query dispatching and result aggregation. Indeed, nugget retrieval has been the research target for more than 20 years from now. How many results from each vertical to return and where to position them in the result page?

--- Slotting results according to the context.

--- Users looking at few result pages. What about remaining pages?

--- Should there be any limit on the retrieved document?

There should be proper evaluation technique to give the relevant score for the summarized content of the particular link .i.e. displayed summary is relevant or not [14].The displayed summary can include multimedia content too in it. Abstractive summarization should be further explored to match the human intelligence and give the accurate and much relevant summary about the content.

To conclude, we believe that this survey in conjunction with other ongoing research indicate that future IR can integrate more focus, structure, and semantics in search results.

REFERENCES

[1] Jamie Callan.Distributed information retrieval: Advance in Information Retrieval, *W. Bruce Croft* (Ed.) Kluwer Academic Publishers, Dordrecht, 235–266, 2000.

[2] Milad Shokouhi and Luo Si. Federated Search. Foundations and Trends in Information Retrieval Vol.5, No.1 (2001) 1-102.

[3] Pal Aditya and Kawale Jaya. 2008. Leveraging query association in federated search. In Proc. of SIGIR 2008, Workshop on Aggregated Search.

[4] Jaime Arguello, Fernando Diaz, and Jean-François Paiement.Vertical selection in the presence of unlabeled verticals. In *Proc. of SIGIR 2010. 691–69,.2000*

[5] Jaime Arguello, Fernando Diaz, Jamie Callan. Learning to Aggregate Vertical Results into Web Search Results. In CIKM' 11.

[6] Kopliku, Arlind and Pinel-Sauvagnat, Karen and Boughanem, Mohand. Aggregated search: a new information retrieval paradigm. (2014) ACM Computing Surveys, vol. 46 (n° 3). pp. 1-31. ISSN 0360-0300.

[7] Christopher Manning, Prabhakar Raghavan, and Heinrich Schutze. Introduction to Information Retrieval. Cambridge University Press.2008.

[8] Jaime Arguello, Fernando Diaz, Jamie Callan, and Ben Carterette. A methodology for evaluating aggregated search results. In Proc. of ECIR 2011. 141–152, 2011.

[9] Yi Guo and George Stylios, "An Intelligent Algorithm for Automatic Document Summarization", Heriot-Watt University, Scotland. 2003.

[10] Mandar Mitra, Amit Singhal and Chris Buckley, "Automatic Text Summarization by Paragraph Extraction", Cornell University.

[11] Wesley T. Chaung and Jihoon Yang, "Extracting Sentence Segments for Text Summarization: A Machine Learning approach", CSE Deapartment, UCLA, Los Angeles, USA.

[12] Giriprasad Sridhara, Emily Hill, Divya Muppaneni, Lori Pollock and K.Vijay Shankar. Towards Automatically Generating Summary Comments for Java Methods.Dept. Of Computer and Information Sciences, University of Delaware Newark, USA.

[13] E. Filatova and V. Hatzivassiloglou, "Event-based extractive summarization," in Proceedings of ACL Workshop on Summarization, vol. 111, 2004

[14] E. Hovy, C. Lin, L. Zhou, and J. Fukumoto, "Automated summarization evaluation with basic elements," in Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC). Citeseer, 2006, pp. 899–902.