



Hybrid Method for Framing Abstractive Summaries of Tweets

Haritha Akkineni^{1*}, Venkata Swarajya Lakshmi Papineni², Vijaya Babu Burra³

^{1,3}Computer Science and Engineering, KL University, Guntur, Andhra Pradesh, India

^{1,2}PVP Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

* Corresponding author's Email: akkinenih@gmail.com

Abstract: Ubiquitous online communication is producing massive amounts of data on an un-precedential scale. Many have come up in analyzing the data produced from Twitter. Sentiment analysis is being carried out to catch the pulse of the people. As the data is mostly in the unstructured format. Getting the summary of what has been expressed positively and what has been expressed negatively plays a major role. Summaries take a wide strand in this plait as the relevant content can't be comprehended all at once. Working on the premise that online social media conversations might represent a new source of information to monitor the status of the policies launched by the government, this investigate stands as a first paw which may ultimately be the source for producing abstractive summaries from Twitter. This paper mainly focuses on developing a hybrid method which takes the combination of extractive summaries using statistical approaches as well as abstractive summaries using Rich Semantic Graph based approaches. The set of classified positive and the negative tweets will be passed as input to the hybrid method and it generates concise as well as readable summaries from both set of tweets. These summaries can be easily viewable on the PDA's and enhances the decision making capability of the policymaker.

Keywords: Abstractive summary, Extractive summary, Syntactic parsing, Morpheme.

1. Introduction

Information overload is a problem in modern digital society caused by the explosion of the amount of information produced on the social networking sites like twitter. A well written summary can significantly reduce the amount of cognitive work needed to digest large amounts of text. Twitter has got abundant of data on the policies launched by the government. These are sentiment classified. These sentiment classified documents are massive and difficult to interpret. This leads to the advent of usage of Text Summarization for the classified tweets. The information which we are going to deal with is at most crucial and will be used by the policy makers in changing or making the decision. They have to be presented to the end user in a more precise and concise manner giving only the essence of what is being presented. Extractive summary identifies important sections of the text and producing them exactly while abstractive

summary aims to produce important material in a new generalized form. Abstraction uses linguistic method to interpret and examine the text. It will produce a generalized summary. It helps the end users to manage the vast amount of information available, by condensing the document content and extracting the most relevant facts or topics included in them. Abstractive summarization requires deep understanding and reasoning over the text, determining the explicit or implicit meaning of each element, such as words, phrases, sentences and paragraphs, and making inferences about their properties in order to generate new sentences which compose the summary [1]. In extractive summaries, Important sentences are excluded from the summary just because of the fact that their frequency does not satisfy a threshold value. Extractive summarization have been studied for years but not suitable for generating concise summaries and summarizing highly redundant text. Bias with limit on the summary size, May contain irrelevant information, Not suited for smaller devices.

Ideal Summary should contain the important information summarized and concise and readable. In Twitter after sentiment mining is done the abstractive summarization aims is to create a compact version of the source text and preserves the important information. In this paper, an approach is presented to generate an abstractive summary for the input document containing a set of positive tweets and documents containing the set of negative tweets. It uses both statistical analysis and rich semantic graph based techniques to generate abstractive summary of the given content. We are proposing a more appropriate strategy would be to combine extractive and abstractive information, improving the performance of the resulting summaries considerably. This paper proposes a system that accepts a document as input and processes the input by using probabilistic function to generate most frequently used tweets and this output will be the input for second phase where linguistic processing and a rich semantic graph construction of the original document is done. A sub graph selection is made using heuristic rules. Finally natural language generation is done which results in automatic generation of the summary. The current systems were able to produce structured formats which are useful but not enough. The star rating was the traditional method in use. To know more the user had to read more sentences. Our system gives a supporting textual summary. It summarizes the major opinions which are concise and are readable. Related work and literature survey is discussed in section 2. The architecture of proposed system is discussed in section 4 and 5. Section 6 discusses on the evaluation metrics taking various other methods, future work is given in 7 and conclusion in section 8.

2. Literature review

A single document summarizers Summarist, is created based on the 'equation': summarization = topic identification + interpretation + generation. It is based on the understanding of whole text and re-phrasing it in fewer sentences. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document.[2]. This study focuses on learning models for extractive summaries from text. It seems most natural to try to model human abstracting. In this study we focus on learning models for extracting summaries from document text. They applied machine learning algorithms to capture characteristics of human extracted summary

sentences. It cannot perform concept interpretation, as it had to incorporate more elaborated concept taxonomy than it currently has.

The approach proposed in [3] automatically fuse similar sentences across news articles on the same event. The method uses language generation for producing concise summary. In this approach, first the similar sentences are pre-processed using a shallow parser and then sentences are mapped to predicate-argument structure. Next, the content planner uses theme intersection algorithm to determine common phrases by comparing the predicate-argument structures. Those phrases that convey common information are selected and ordered and some information are also added with it(temporal references, entity descriptions). Finally sentence generation phase uses FUF/SURGE language generator to combine and arrange the selected phrases into new summary sentences. The major strength of this approach is that the use of language generator significantly improved the quality of resultant summaries i.e. reducing repetitions and increasing fluency. The problem with this approach is that context of sentence was not included while capturing the intersected phrase.

A clustered semantic graph based approach for multi-document abstractive summarization was proposed. The approach operates by employing semantic role labelling (SRL) to extract the semantic structure (predicate argument structures) from the document text. The predicate argument structures (PASs) are compared pair wise based on Lin semantic similarity measure to build semantic similarity matrix, which is thus represented as semantic graph whereas the vertices of graph represent the PASs and the edges correspond to the semantic similarity weight between the vertices. Content selection for summary is made by ranking the important graph vertices (PASs) based on modified graph based ranking algorithm. Agglomerative hierarchical clustering is performed to eliminate redundancy in such a way that representative PAS with the highest salience score from each cluster is chosen, and fed to language generation to generate summary sentences. Experiment of this study is performed using DUC-2002, a standard corpus for text summarization. Experimental results reveal that the proposed approach outperforms other summarization systems [4].

I. Fathy, D. Fadl, M. Aref [5] proposed a new semantic representation called Rich Semantic Graph (RSG) to be used as an intermediate representation for various applications. A new model to generate an English text from RSG is proposed. The method

access a domain ontology which contains the information needed in same domain of RSG. Coreference resolution was not addressed in this paper to remove the duplications

3. Algorithm

Input - I. Text Data for which Summary is required.

Output - O. Summary for the Original Text Data.

Step 1 Accept input document.

Step 2 Pre-processing the positive tweets.

- a) Separate all the tweets on the basis of punctuation marks and record in array S_j .
- b) Separate all the words on the basis of space, punctuations etc.
- c) Eliminate all the stop words

Step 3 All the derivations of the words are replaced by its root word

Step 4 Calculate the probability of each word (w_i) occurring in the input document:

$$p(w) = n/N \quad (1)$$

In Eq.(1), n denotes for frequency of particular word N denotes total number of words

Step 5 Replace semantically similar words by their root words and update frequencies.

Step 6 Assign Weight to each sentence.

Equation (2) gives the weight assigned to each sentence is equal to the average of the probability of all the words assigned in step 1.

$$Weight(S_j) = \sum_{w_i \in S_j} p(w_i) / \{ |w_i| / w_i \in S_j \} \quad (2)$$

Step 7 Pick out the sentences in descending order of their probabilities. Output the text document

Step 8 Create a rich Semantic graph from the text document.

Step 9 Reduce the graph to a more abstracted graph

Step 10 Generate a abstractive summary.

4. Explanation of the architecture

Extractive summarizer block:

In the pre processing phase tweets are separated by punctuation marks like comma, semi colon, full stop etc.

The words are separated by blank spaces, and stop word elimination like helping verbs, articles etc. Stemming of the remaining document is done where all the derivations of words are converted to its root word using Porter Stemming algorithm [6]. The frequency of each word is calculated by considering semantically similar words as well.

Weights are assigned to each word on the basis of probability given to each word on the basis of formula mentioned below in algorithm. Then the sentence containing maximum number of words with higher frequency is chosen to give the maximum weight and the same procedure is followed with rest of the sentences. Ranking of each sentence is done using the probability assigned to each word and further the weight of sentence which will be done. The sentences will be taken to the text document and will be the input to the Abstractive summarizer block.

Abstractive summarizer block:

This block is a combination of three phases.

1. Rich Semantic Graph creation
2. Graph reduction
3. Generate abstractive summaries.

Rich semantic graph creation:

The Objective of this phase is to represent the given Tweet document semantically using RSG. It consists of 3 steps like 1. Pre-processing 2. Rich semantic sub graph generation 3. Rich Semantic graph generation.

Pre-processing consists of four main processes:

1. named entity recognition: This process locates atomic elements into predefined categories such as person names, organizations, etc. SENNA - Software is used for performing Named Entity Recognition. Stanford Named Entity recognition can be imported.

The named entity recognition process:

Example: ["Jim", "PERSON"],
["went", "O"],
["to", "O"],
["Stanford", "ORGANIZATION"],
["University", "ORGANIZATION"]

The NER obtained from Stanford Named Entity Recognition.

Morphological and syntactic analysis:

The study of the word formation is morphology. Each word is divided into morphemes and figures out its grammatical categories. It helps in identifying, analyzing and describing the structure of the given language. MORPHEMES and other linguistic units such as root words, suffixes, POS stress or implied content.

Example:
Washing=Wash+ing
LEMMA:Walking
Part of speech: noun

Grammar: narrative,singular
forms: walking,walkings

Syntactic analysis:

Each words syntactic function is described and parse tree is constructed. We used Stanford implementation of POS tagging to achieve our task. This software uses log linear POS tagger. Following is an example to illustrate its use:

Input text:

"Mary was happy for the help provided by John".
The output of Stanford POS tagger is a parse tree (ROOT

```
(S
  (NP (NNP Mary))
  (VP (VBD was)
    (ADJP (JJ happy)
      (PP (IN for)
        (NP
          (NP (DT the) (NN help))
          (VP (VBN provided)
            (PP (IN by)
              (NP (NNP John))))))
        (. )))
```

The parse tree will help us to find nouns and verbs in the sentence.[7]

Co-reference resolution:

Consolidating expressions that refer to the same named entity can be done using co reference resolution. This can be done using surface form matching and text layout analysis. [8]

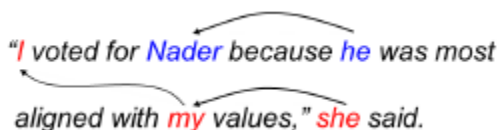


Figure.2 Co reference resolution

In the example, "Nader" and "he" are members of one cluster, and "I," "my," and "she" are members of another. Due to the diversity of surface forms for linguistic reference and special phenomena such as pronominal co reference.

Pronominal resolution reference

It is use to form chains in the documents and resolve the pronouns with the respective subjects. Replacing pronouns with their respective nouns will help us in the later phrases where measurement of context or flow of information is difficult to manage. The objective of this phase is to get rid of ambiguity.

We then use NLPW in syntactic and semantic tags to trace and resolve pronominal references[9] as they appear in the text.

Example:

input: "John is a musician. He played a new song. A girl was listening to the song."

```
[
  {"John": ["a musician", "He"]},
  {"a new song": ["the song"]}
]
```

Figure.3 Pronominal resolution

Syntatic Analyser tool: Lingsoft

Parser tool: Stanford University Natural Language group.

These are used for pronominal resolution. This is how a rich semantic graph of tweets is created.

Semantic sub graph generation

In this stage multiple rich semantic sub graphs are generated for each and every processed sentences. The sentences are of the form $S_i = [W_{i1}, W_{i2}, \dots, W_{in}]$, where W_{ij} is a word j belonging to a sentence i . Each word is represented as a triple sequence $W_{ij} = [St, T, D]$. St denotes the word stem, T represents morphological and syntactic tags and D denotes the set of typed dependency relations. This module includes three processes: Word Senses Instantiation, Concepts Validation, and Semantic Sentences Ranking processes.

Word senses instantiation:

The different word senses for noun and verb are formed based on domain ontology.

For example, Input : Student

These senses are gathered using WordNet Search
Output:

```
Noun
• S: (n) student, pupil, educatee (a learner who is enrolled in an educational institution)
• S: (n) scholar, scholarly person, bookman, student (a learned person (especially in the humanities); someone who by long study has gained mastery in one or more disciplines)
```

Figure.4 Word sense instantiation

Concept validation:

Which of the sentences generated leads to a valid concept have to be recognized in this stage. For example, the "student" noun has two concepts: the first concept is a kind of enrollee or pupil, and the second is equivalent to scholar. Therefore, the phrase "graduate student" is valid only with the first

concept. This deviation of the concept have to be validated.

Sentences ranking:

It aims to rank and to threshold the highest ranked rich semantic sub-graphs for each sentence. To generate single rich semantic graph and to keep the semantic consistency for the whole sentence, the process considers the first ranked rich semantic sub-graph only.

The rich semantic graph generation module

Finally, the Rich Semantic Graph Generation module is responsible to generate the final rich semantic graphs of the whole input document from the highest-ranked rich semantic sub-graphs of the document sentences. The semantic sub-graphs of the input document will be merged to form the final rich semantic graph. A sample of creation of semantic graph [10].

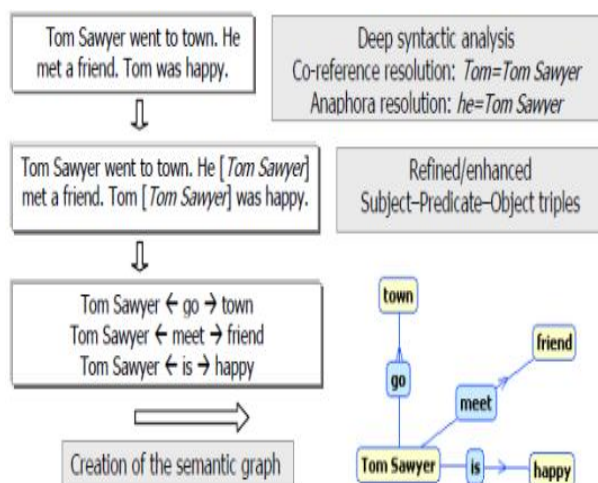


Figure.5 Semantic graph generation

Rich semantic graph reduction

The Original document is converted to more reduced graph. In this phase, a set of heuristic rules are applied on the generated rich semantic graph to reduce it by merging, deleting, or consolidating the graph nodes. These rules exploit the Word Net semantic relations: hypernym, holonym, and entailment. There are many rules can be derived based on many factors: the semantic relation, the graph node type (noun or verb), the similarity or dissimilarity between graph nodes, etc.

Table 1 presents a set of heuristic rule examples that can be applied on the graph nodes of two simple sentences:

- Sen1=[SN1,MV1,ON1] and
- Sen2=[SN2,MV2,ON2].

Table 1. Different summarization techniques have been evaluated by calculating the compression ratio and readability test.

Summarizer	Number of words in Text Data	Number of words in Summary	Percentage
Automatic Text summarizer[16]	401	70	17.4
Free Summarizer[17]	401	165	41.14
Text Compactor[18]	401	65	16.20
Summary() in R[19]	401	89	22.19
Tools4Knoobs[20]	401	181	45
Proposed Method-Hybrid Method	401	90	22.44

Each sentence is composed of three nodes: Subject Noun (SN) node, Main Verb (MV) node, and Object Noun (ON) node. For example, in rule 1, both main verbs (MV1 and MV2) are merged and both sentence objects (ON1 and ON2) are merged if the two sentence subjects are instances of the same noun (N), both sentence verbs are similar, and finally both sentence objects are similar [11]. Some of the rules are:

Heuristic rules

Rule 1:

*IF SN1 is instance of noun N And
 SN2 is instance of noun N And
 MV1 is similar to MV2 And
 ON1 is similar to ON2
 THEN Merge both MV1 and MV2 And
 Merge both ON1 and ON2*

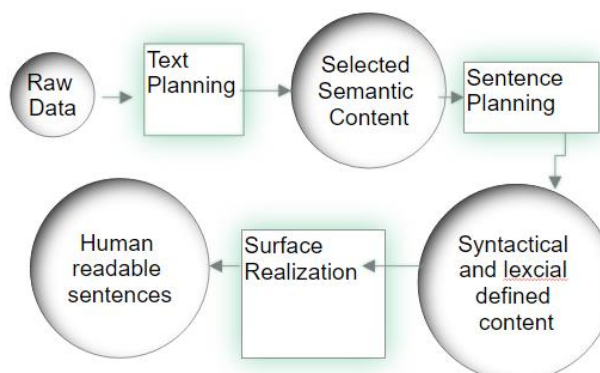


Figure.6 The standard pipeline architecture for natural language generation

The summarized text generation phase

This phase aims to generate the abstractive summary from the reduced Rich Semantic Graph (RSG). To achieve its task, the phase accesses the domain ontology, which contains the information needed in the same domain of RSG to generate the final texts [12].

The Text planning phase

It decides what information should be included in the generated text.

The Sentence Planning phase

It improves the fluency or understand ability of the text. To achieve this objective, the words of the text should be related to each other, the clauses should exhibit no unintentional redundancy, and the different sentences with the same subject should be aggregated. The sentence planning consists of four main processes: Lexicalization, Discourse Structuring, Aggregation, and Referring Expression processes. Decomposing language this way allows us to formulate the probability of a sentence in terms of the probabilities of each word using the chain rule:

$$P(S)=P(w_1).P(w_2/w_1).P(w_3/w_{1,w_2}....) \quad (3)$$

Surface realization phase

This module aims to transform the enhanced semi- paragraphs into paragraphs by correcting them grammatically and adding the required punctuation. The techniques of SimpleNlg (Simple natural language generation) [13] can be exploited to achieve these objectives.

The evaluation phase

The main objective of this module is to evaluate and then rank the paragraphs according to the coherence between paragraph sentences. Coherence in a paragraph is the technique of making words, phrases, and sentences move smoothly and logically from one to the other. In other words, the ideas are so interwoven and "glued" together that the reader will be able to see the consistent relationship between them.

Firstly, text coherence evaluation is applied for assessing whether the paragraphs are coherent or not. Therefore, each paragraph is evaluated and ranked according to the number of coherence relations [14] between its sentences. Finally, the final paragraphs can be sorted according to the coherence evaluation rank [15]. After going through all these modules we finally get an abstracted version of the given tweets.

5. Experimental results

Compression Ratio: The compression ratio is calculated as the Length of Summary/Length of text. It denotes the ratio of the reduction of the text within the original document versus the summarized one. The main intention of framing summary is to make the enormous content in a precise form which can be proved by calculating the compression ratio. The compression ratio is calculated for all the above methods and the hybrid method proposed could give a moderate compression ratio not too high or too low. This itself shows that our hybrid method moderately achieves our aim. Different online available tools and different text packages were used to evaluate this factor (the compression ratio) and we ended up in the above table.

Readability Test: To test the readability, we conducted a readability test using summaries generated from Hybrid Method. We used a human evaluator to work with this process. He was asked to pick the 2 least readable sentences from each of the 30 test sets (based on 30 summaries). Collectively, there were 339 sentences out of which 62 were Hybrid-Method generated. Out of these, the human assessor picked only 11 of the sentences as being least readable, resulting in an average readability score of 0.51. This shows that more than 50% of the generated sentences are indistinguishable from human composed sentences. Of the 11 sentences with problems, 5 contained no information or were incomprehensible, 3 were incomplete when sentence validity check was done, and 3 had conflicting information due to mixed feelings about the topic. We use multiple reference (human) summaries in our evaluation since it can achieve better correlation with human judgment. We had taken 10 different human workers to summarize each review document. The workers were asked to be concise and were asked to summarize the major opinions in the review document presented to them. We manually reviewed each set of reference summaries and dropped summaries that had little or no correlation with the majority. This left us with around 4 reference summaries for each review document. We introduce a readability test to understand if Hybrid Method summaries are in fact readable. Suppose we have N sentences from system-generated summary and M sentences from corresponding human summaries. We mix all these sentences and then ask a human assessor to pick at most N sentences that are least readable as the prediction of system summary.

$$\text{Readability}(O) = 1 - \#CorrectPick/N \quad (4)$$

Table 2. Summary quality attributes

Summary Quality Attributes			
	Quality	Mean	Standard Deviation
1.	Information Content	4.12	0.98
2.	Grammatical Correctness	3.70	0.86
3.	Abstractness	3.42	1.30
4.	Expressiveness	3.71	0.84
5.	Unnecessary information	3.25	1.56

If the human assessor often picks out system generated summaries as being least readable, then the readability of system summaries is poor. If not, then the system generated summaries are no different from human summaries. So according to the analysis done by us the hybrid method resulted in the average readability score of 0.51%. It shows that it achieves better readability when compared to the other methods. As readability is the main measure to be considered with respect to framing summaries.

To measure the qualities of a summary: Like expressiveness, information content, abstraction we performed a human evaluation. We executed our system on some schemes. We had summaries on those schemes beforehand. I asked humans to rate our system generated summaries in reference to human compiled respective summaries, on the grounds of Information Content, Grammatical Correctness, Abstraction, Expressiveness and Excess or Unnecessary Information. These qualities are rated on a scale of 1 to 5, 1 being the least and 5 being the most. The mean and standard deviation observed over these qualities is depicted in the table. The results suggest that information content is good in our summaries generated. But some people are of the opinion that there is a lot of unrelated information, and the standard deviation varies a lot. Also we need to improve upon the abstractness quality, as most of the frames are just a short hand representation of sentences in document and hence turn out to be factual. The proposed technique takes the complete blend of extractive as well as abstractive summaries and generates the abstractive summary taking the main essence of both these methods. Whereas the existing techniques concentrated on either extractive or abstractive versions alone. The method proposed reduces time spent by researchers, decision makers etc. by access to quality abstracts of sentimental tweets. Tweets on Digital India were extracted and the abstractive summaries were generated using the hybrid system.



Figure.7 Input

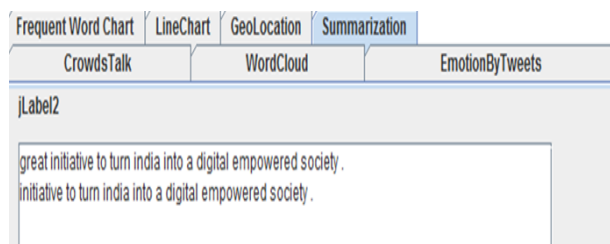


Figure.8 Abstractive summarized content as output

6. Conclusion

The algorithm which was proposed is useful in generating abstractive summaries for the given set of positive tweets as well as the negative tweets. This helps the end users to get concise and precise information. It assist them to view it on the smaller screens as well in order to augment their decision making process. This method facilitates them to have an access to relevant and important facts immediately. Humans have the tendency to oversee important and critical facts or sentences. However the summarizer framed using hybrid method will automatically cover the important facts of the document.

On par with the other summarization techniques the hybrid method proposed could perform well in the compression ratio calculations. The readability is really ahead of the remaining techniques. The results showed that the information content in the summary generated is sufficient. The proposed method was able to achieve an average compression ratio of 22.44. The readability it has achieved is more than 50%. It shows that the summaries are readable. The summary quality attributes. The information content was high which resulted from a mean of 4.12. It shows that the summary generated was informative.

7. Future scope

We can conclude that this hybrid method goes well if the document is fed as a input for the algorithm. We have to research on if this method work for query focus summarization. How this information could be generalized to form a concept map is still a challenging task.

References

- [1] P. Norvig, "Inference in text understanding", In: *Proc. of the Sixth National Conf. on Artificial Intelligence*, Seattle, Washington, pp.2561-2565, 1987.
- [2] E. Hovy and C.Y. Lin, "Automated Text Summarization in SUMMARIST", In: *Proc. of ACL 97/EACL 97 Workshop on Intelligent, Scalable Text Summarization*, pp. 66-73, 1999.
- [3] R. Barzilay, K. McKeown, and M. Elhadad, "Information fusion in the context of multi-document summarization", In: *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 550- 557, 1999.
- [4] A. Khan, N. Salima, W. Reafeea, A. Sukpraserta, and Y. J. Kumarb, "A clustered semantic graph approach for multi-document abstractive summarization", *Jurnal Teknologi (Sciences & Engineering)*, Vol. 77, No. 18, pp. 61–72, 2015.
- [5] I. Fathy, D. Fadl, and M. Aref, "Rich Semantic Representation Based Approach for Text Generation", In: *Proc. of the 8th International Conf. on Informatics and Systems*, pp. 552-560, 2012.
- [6] C. Moral, A. Antonio, R. Imbert, and J. Ramírez, "A Survey of Stemming Algorithms in Information Retrieval", *Information Research*, Vol. 19, No. 1, 2014.
- [7] D. Bhartiya and A. Singh, "A Semantic Approach to Summarization", *arXiv:1406.1203*, 2014.
- [8] X. Cheng and R. Voigt, "A Deep Architecture for Coreference Resolution", In: *Proc. of the 2001 Workshop on Computational Natural Language*, pp. 1-8, 2015.
- [9] J. Leskovec, M. Grobelnik, and N. Frayling, "Learning semantic graph mapping for document summarization", In: *Proc. of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies*, 2004.
- [10] J. Leskovec, N. Frayling, and M. Grobelnik, "Extracting Summary Sentences Based on the Document Semantic Graph", *Microsoft Research Technical Report, No. MSR-TR-2005-07*, 2005.
- [11] D. Bartakke, S. Sawarkar, and A. Gulati, "A Semantic Based Approach for Abstractive Multi-Document Text Summarization", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 4, No. 7, pp.13620-13629, 2016.
- [12] T. Packer, "Surface Realization Using a Featurized Syntactic Statistical Language Model", *All Theses and Dissertations*, pp.1-94, 2006.
- [13] R. Barzilay and M. Elhadad, "Using Lexical Chains for Text Summarization", In: *Proc. of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 10–17, 1997.
- [14] F. Wolf and E. Gibson, "Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance", In: *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp. 383–390, 2004.
- [15] I. Moawad and M. Aref, "Semantic Graph Reduction Approach for Abstractive Text Summarization", In: *Proc. of the International Conference on Computer Engineering & Systems*, pp. 132–138, 2012.
- [16] Automatic Text Summarizer, Retrieved from <http://autosummarizer.com>, 2013
- [17] Free Summarizer, Retrieved from <http://freesummarizer.com/>
- [18] K. Edyburn, Text Compactor. Retrieved from <http://textcompactor.com>, 2010
- [19] R Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <http://www.R-project.org/>, 2013
- [20] Online summarize tool (free summarizing), Retrieved from <https://www.tools4noobs.com/summarize>, 2007