2015 International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, Mumbai, India

Automatic Text Summarization of Wikipedia Articles

Dharmendra Hingu[1], Deep Shah[2], Sandeep S. Udmale[3]
Department of Computer Engineering and Information Technology
Veermata Jijabai Technological Institute
Mumbai, India
[1]dharmendra.hingu@gmail.com, [2]deepshah@live.com, [3]sandeep_udmale1@yahoo.co.in

***Abstract*—The main objective of a text summarization system is to identify the most important information from the given text and present it to the end users. In this paper, Wikipedia articles are given as input to system and extractive text summarization is presented by identifying text features and scoring the sentences accordingly. The text is first pre-processed to tokenize the sentences and perform stemming operations. We then score the sentences using the different text features. Two novel approaches implemented are using the citations present in the text and identifying synonyms. These features along with the traditional methods are used to score the sentences. The scores are used to classify the sentence to be in the summary text or not with the help of a neural network. The user can provide what percentage of the original text should be in the summary. It is found that scoring the sentences based on citations gives the best results.**

Keywords—Natural Language, Text summarization, Python, Frequency

I. INTRODUCTION

Natural languages can be broadly defined as any informal language that can be learnt by any person. They differ from formal languages like computer programming languages which have a proper structure and syntax. Natural Language Processing (NLP) is the understanding of the context of these natural languages and/or generating a text in natural language. Radev et al. defines a *summary* as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that" [1].

The objective of automatic text summarization (automatic summarization/text summarization) is the reduction of a given text to a smaller number of sentences without leaving out the main ideas of the original text [2].

The architecture of a summarization system consists of interpretation, transformation and generation. Text summarization is broadly classified into four categories. The first category is a *classical approach* which relies on a selection of salient information based on features like frequency, cue phrases, title and heading words, and sentence location [3]. The second category is a *corpus-based approach* which uses the concept of statistics and then decides the appropriate combination of features mentioned in classical approach [3]. The third category is an *exploiting discourse structure* which exploits models of discourse structure, while maintaining a relatively domain-independent and knowledge poor approach [3]. The final category is *knowledge rich approach* which models the rich knowledge requirements for summarization in a particular domain. In this category, a relatively rich structured representation has been built, which the summarization process used as an input. First three categories focus mainly on interpretation but final category focuses on transformation and generation [3].

Microsoft word 2007 uses the summarization algorithm which identifies word frequency in the given documents and a percentage of higher scored sentences are displayed in the summary.

II. PROPOSED SYSTEM

The proposed system solely focuses on extractive based summarization method. We will discuss two new approaches for summarizing the text. The first method is to adjust the frequency of the words based on the root form of the word, and also the frequency of its synonyms present in the text. The second method is to identify sentences containing citations or references and give them a higher weight. As shown in Fig. 1, the summarization system takes input as Wikipedia articles, processes it and gives the summary sentences. Input file consist of raw data to be processed by the system.

The first step of the summarization is known as tokenization, which break down the passages into sentences and each of these sentences is further broken into a set of words. The data structure obtained after tokenization of passages is a list, containing each element as a sentence and data structure obtained after tokenization of sentences is a list of list, containing sets of words. Tokenization is performed through pattern matching using 'regular expressions'. Data obtained in the form of set of words is further analyzed and stop words or most commonly occurring words, like articles, are removed from the set of words.

The words are then stemmed, that is brought to their root form. The main objective is to assign equal importance to words having the same root. Thus, words in their different forms are considered to be the same. For e.g. the words likes 'compute', 'computed', 'computing', 'computer', 'computation', and 'computable' are brought to the root form 'comput'. Further, the text may have synonyms, it is essential to assign the same weight to words having the same meaning. Hence, synonym checking is done. Only certain features require stemming and synonym checking.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

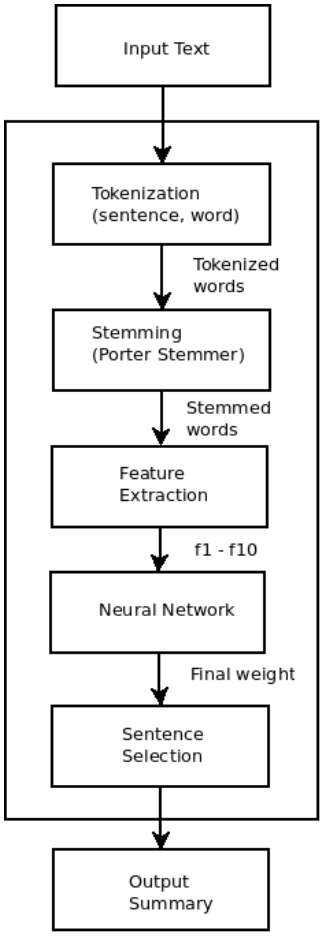

Fig. 1: Text Summarization System

Next step is feature extraction. The proposed automatic text summarization system consists of scoring sentences based on multiple features. The features extraction criteria for any sentences *s* are

1) Relative position of sentence [9]: Sentences at the start of the passage (Introductory) and at the end (Conclusive) are usually important. These sentences will be given additional weights ranging from one to five, while the remaining sentence will have a weight of zero.

$$f1(s) = \frac{Assigned\ \ position\ value}{Total\ importance} \quad (1)$$

2) Named entities [9]: Weights will be assigned to sentences containing named entities (Proper Nouns), since named entities usually contain key information.

$$f2(s) = number\ of\ named\ entities\ in\ s \quad (2)$$

3) Similarities with other sentences (Bushy path method) [9]: This value signifies how well a sentence is linked to other sentences in the document. This is calculated by counting the common words between the two sentences and dividing it by the length of the longer one. Thus, a graph is created with these values as edge weights and the sentences as nodes. Edges below a certain threshold are removed. The weight of the sentence is the number of edges. Thus, sentences, which have very little relevance with each other, are not connected in the graph.

$$f3(s) = Number\ of\ edges\ connected\ to\ node\ corresponding\ to\ s \quad (3)$$

4) Similarity with rest of the document [9]: This feature shows how much a sentence is related to the entire document. Here the words in the sentence, which match, with other words in the rest of the document are counted and divided by the total words in the document. Thus, the relevance of a sentence to the theme of the document is given by this feature.

$$f4(s) = \frac{Keywords\ in\ s\ and\ keywords\ in\ other\ sentences}{Keywords\ in\ s\ or\ keywords\ in\ other\ sentences} \quad (4)$$

5) Similarities with other sentences (Aggregate similarities) [9]: Aggregate similarity is the sum of all the edge weights in the bushy path method. Even the edges having weights below the threshold will be considered. This feature will take into account how well the two sentences are connected.

$$f5(s) = Sum\ of\ all\ edge\ weights\ of\ a\ node\ corresponding\ to\ s \quad (5)$$

6) Title relevance [9]: The title can be used to find the key sentences in the passage. The number of common words between each sentence and the title are counted and then divided by the total number of unique words in the title and the respective sentences.

$$f6(s) = \frac{Keywords\ in\ s\ and\ keywords\ in\ title}{Keywords\ in\ s\ or\ keywords\ in\ title} \quad (6)$$

7) Relative length of sentence [9]: A longer sentence will tend to contain more information while a very short one may contain no information at all. Thus, this feature measures the sentences based on their lengths.

$$f7(s) = Average\ sentence\ length * length(s) \quad (7)$$

8) Frequency of words: The frequency of each word in the entire document is calculated. Then the weight of each sentence is calculated by summing the frequency of all the words in the sentence and then divided by the length of the sentence. Synonyms and words in the same root form have same weight. Term frequency-inverse document frequency (TF-IDF) is used in this feature.

$$f8(s) = \text{Sum of frequency of words in s} - \text{Frequency of stop words in s} \quad (8)$$

9) Citation: Wikipedia or other documents might have a citation (references) to the words or the sentences. Usually these words or sentences are important. A weight of one is assigned to the sentences having citations, zero otherwise.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

$$f9(s) = \begin{cases} 1 : If\ s\ has\ citations, \\ 0 : Otherwise \end{cases} \quad (9)$$

10) Numerical data [9]: If at all, any numerical data is available in the document, they are important. Hence, a weight of one is assigned to the sentences having numerical values, zero otherwise.

$$f10(s) = \begin{cases} 1 : If\ s\ has\ numerical\ value, \\ 0 : Otherwise \end{cases} \quad (10)$$

These scores are then fed to a neural network, giving a single output score. Both, the input feature scores and the output score have a range from zero to one. Neural network produces a single value, signifying the importance of the sentence in the summary.

III. Experimental Results

The objective is to build a system that could summarize a text when an input is given as a Wikipedia article to it. For the development of the system, Python's NLTK toolkit, porter stemmer algorithm and ffnet library are used. NLTK toolkit is a text processing tool. Porter stemmer is used to define a set of rules to find out the root of the word [10]. The ffnet library allows us to use neural network algorithms. NLTK toolkit and ffnet neural network model aids for platform independence.

As shown in fig. 2, the entire summarization system consists of the following stages- The input to the system is a text file (any type of text), which is given to the tokenization modules to generate sets of words. Using the porter stemmer algorithm, these sets of words are brought to their root forms. Further, features are extracted for the given input. After extracting the features, they are normalized to range from zero to one. For each sentence, these weights are then fed to a feed-forward neural network. The network will thus contain 10 input nodes, 1 hidden layer and 1 output node. Supervised learning will be used to train the network. Both, the input feature scores and the output score have a range from zero to one. The score is directly proportional to the importance of the sentence. These scores are then used to generate the summary.

Results of the system are evaluated using the *F1* score. *F1* score or F-measure is used to calculate the accuracy of a certain system. *F1* score calculation uses both, precision (p) and recall (r). Precision is the fraction of the summary that is correct. Recall is the fraction of the correct (model) summary that is outputted. Recall measures the sensitivity of the system. *F1* score is a combined score (harmonic mean) which considers both, the precision and the recall. Precision and recall are calculated based on the true positives, false positives and false negatives of the results found. True positive (tp) is a result, which is found to be true, and is actually true. False positive (fp) is a result, which is found to be true but is actually false. False negative (fn) is a result, which is found to be false but is actually true.

$$Precision = \frac{tp}{tp + fp} \quad (11)$$

$$Recall = \frac{tp}{tp + fn} \quad (12)$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (13)$$

The system is evaluated with fifteen summaries. Summaries created from Microsoft Word 2007 have been used as model summaries. The precision, recall and F1 scores have been calculated for all systems. We have used 21 different systems for each evaluation. The first one is one with only the first feature, the next is the one with all features except the first feature of the system; third system is one with only the second feature, the next one is with all features except the second feature and so on. The last system is one using all features of the system.

As shown in fig. 3, only feature f8 has the highest precision of 0.172 while the one with only feature f7 is the worst with precision 0.061. Systems with only feature f6 and only feature f9 also give high results with precision 0.161 and 0.155 respectively. The system with all the features is moderate with precision 0.15.

As shown in fig. 4, only feature f1 has the best recall of 0.512 while the systems with only f9 and f10 also have a good recall of 0.401 and 0.305 respectively. Again the system with only feature f7 proves to be the worst with a recall of 0.051. The system with all features has a recall of 0.121.

As *F1* score is a combined metric of precision and recall, results of both are reflected in the *F1* scores as shown in fig. 5. The system with only feature f9 is the best with an *F1* score of 0.223 while the systems with only features f1, f6 and f8 have *F1* scores 0.205, 0.153 and 0.152 respectively. As expected, the system with only feature f7 has the lowest *F1* score of 0.055. The system with all the features has an *F1* score of 0.133.

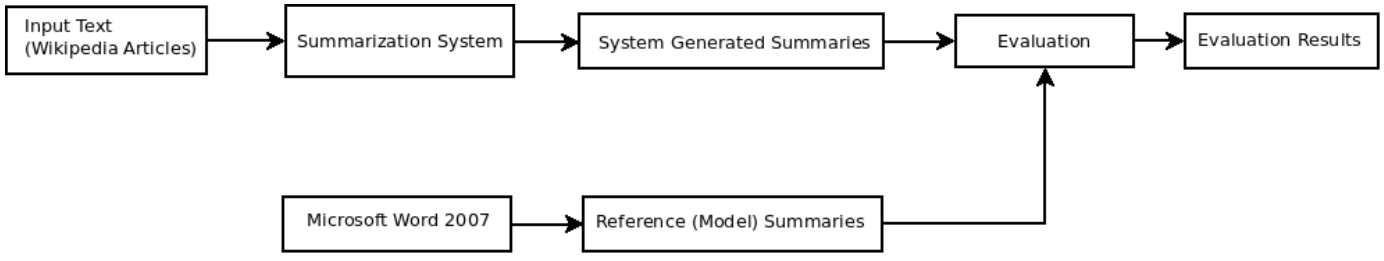

Fig. 2: System Overview

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

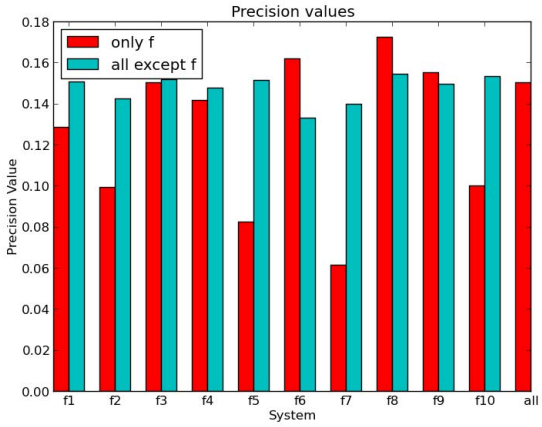

Fig. 3: Precision values

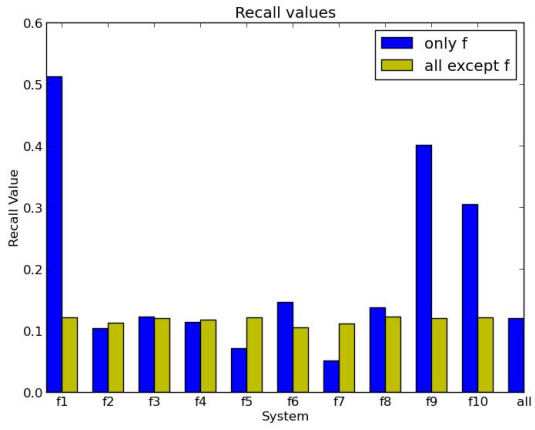

Fig. 4: Recall values

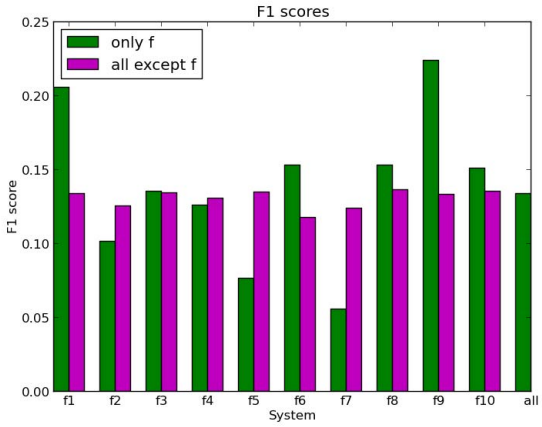

Fig. 5: F1 scores

IV. Conclusion

In the proposed system, equal weights are given to similar words. This includes giving weights to synonyms, and also words having the same roots. Hence, main idea of sentence is considered irrespective of the usage. The system also makes use of citations present in the input document. Sentences with citations are given a higher weight for summarization.

Fig. 3 - 5 indicates that, the systems with only one feature have extreme results while the other systems with all the features, or all the features except one have almost the same results. Best system is the one with only feature f9, which considers citations. Systems with only features f1 (relative position of sentence), f6 (title relevance) and f8 (frequency of words) also have good results. The system with only feature f7 (relative length of sentence) has the worst results.

References

[1] D. R. Radev, E. Hovy, and K. McKeown, ``Introduction to the Special Issue on Summarization," *Computational Linguistics*, vol. 28, no. 4, pp. 399-408, 2002.

[2] Wikipedia, *Automatic summarization*, http://en.wikipedia.org/wiki/Automatic_summarization.

[3] I. Mani and M. T. Maybury, *Advances in Automatic Text Summarization*, MIT Press, 2001.

[4] Elena Lloret, María Teresa Romá-Ferri, and Manuel Palomar, ``COMPENDIUM: A text summarization system for generating abstracts of research papers," *Data \& Knowledge Engineering*, 2013.

[5] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong, ``Comparative Document Summarization via Discriminative Sentence Selection," *ACM Transactions on Knowledge Discovery from Data*, vol. 7, no. 1, March 2013.

[6] M K Dalal and M A Zaveri, ``Heuristics Based Automatic Text Summarization of Unstructured Text," *ACM International Conference and Workshop on Emerging Trends in Technology (ICWET 2011)*, 2011M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] Aminul Islam and Diana Inkpen, ``Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, July 2008.

[8] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng, ``Text summarization using a trainable summarizer and latent semantic analysis," *Information Processing and Management*, vol. 41, pp. 75-95, 2005.

[9] Mohamed Abdel Fattah, and Fuji Ren, ``GA, MR, FFNN, PNN and GMM based models for automatic text summarization," *Computer Speech and Language*, vol. 23, no. 1, pp. 126-144, 2009.

[10] The *Porter Stemming Algorithm*, http://tartarus.org/~martin/PorterStemmer/index.html

978-1-4799-5522-0/15/$31.00 ©2015 IEEE