EUTS: Extractive Urdu Text Summarizer

Noman Jazeb, Sikandar A., Aslam Muhammad
Dept. of CS & Engg., UET, Lahore, Pakistan
mr.jazeb@gmail.com, siki5815@gmail.com, maslam@uet.edu.pk

Abstract—Automatic text summarization is a growing area of natural language processing research. Using extractive text summarization approach a concise summary of the input information sources is developed by selecting phrases and sentences on a given selection criterion that can be based on features e.g. syntactic, semantic, temporal, positional, etc. Text summarization for under resourced languages e.g. Urdu is even more challenging, due to the limited availability of basic computational resources to effectively extract textual features. Surmounting these challenges, this paper presents an extractive text summarization methodology for Urdu language documents based on sentence weight algorithm using segmentation, tokenization and stopwords as prominent features. ROUGE metric is used for system evaluation by comparing system generated and human generated summaries. System accuracy at Unigram, bigram and trigram level is 67 percent.

Keywords— Natural language processing, Automatic text summarization, sentence weight algorithm

I. INTRODUCTION

Nowadays internet turns out to be a strength for every industry especially the information technology. The Internet changes the world as a global village for the spreading of material. Each and every industry uses the Internet for the distribution of information and save that for data processing. This statistics is growing progressively on the intent. Recently, information is emerging so much promptly in the textual form. Exemplifications of documented data are news, articles, blogs and electronic books etc. As tons of data is available on the internet, sometimes it creates difficulty for a person to abstract the maximum substantial information for his particular use [1].

In this modern era, life turns out to be so much swift and everyone does not have so much free time to read out the all raw data very cautiously to abstract the leading subject for his personal satisfaction and save it for imperative time. To get abstract of the significant information from so much raw data, various resources are required. In order to resolve the problem of attaining valuable subject from the bulk of raw data there are numerous solutions accessible according to every arena of science. The roughly universal observation of people prior to watch movies, reading different stuff like books, research documents, novels etc. they look through the previously accessible analysis of above mentioned stuff that are stated by different users to discover the finest stuff for utilization of free time.

So, in our opinion and according to the circumstances explained above, “Summarization” is one of the finest ways for script reading. The summarization of data can definitely provide the most substantial information in less time & resources to discover the most noteworthy content of raw data. In general arguments, the summary can delineate as the detection of conclusion and noiseless features from the briefly defined documents [2]. Text summarization (TS) is one of the dynamic strengths of many scientific areas. TS procedures are solitarily classified into two branches. These two divisions are “Extractive” and “Abstractive” [4].

Extractive summarization commonly abstracts the furthermost information in the given input text or document [5]. Whereas on the other hand, abstractive produces the improved form of imperative information by eradicating the duplication of redundant words and represent them into particular wording as output sentences/documents. Extractive summary fundamentally involves three foremost procedures [3]. The leading development in summarization is to characterize the intermediate text from the original and in following stage rank out the most imperative sentences according to some algorithm. In concluding process, it selects the most applicable and highly marked sentences for the summarization.

Urdu is one of the most speaking languages in sub-continent area. Urdu is very simple and common language as a speaking point of view. Urdu is included in the top 20 languages by the population and speaking statistics all over the world. Urdu is the national language of Pakistan and speaks all over the country more widely. Urdu is basically a mixture of Persian and Arabic language [10]. It writes in right to left manner. One of the best feature or we can say that unique feature that Urdu includes different languages words like English, Hindi, Sanskrit, Persian, Turkish and Arabic etc. Nowadays Urdu get more attention in research to make the systems like voice recognition, search engine, summarization etc. make some progress in Urdu resources for further uses.

The paper is organized in such a way that sections 2 discuss the existing automatic text summarizer for other languages, methods, and algorithms. Section 3 reveals the suggest system complete procedure and functionality. Section 4 illuminates the consequence and argument of recommending the system and in last section conclusion and future work for the system.

II. LITERATURE REVIEW

The language-independent summarization system is absolutely based on universal networking language (UNL). This system assembles multi-language abstracts with the help of inter-language approach for the multi-document representation. This methodology, in the graphical representation, made each and every sentence in the node form. If two findings share same /common words then there are nodes connected by an edge to represent the corresponding of sentences. The significant part of this method is that it recognizes the exact sentence from nodes. For the identifications of imperative sentences and documents, it uses Term Frequency-Inverse Sentence Frequency (TFISF) approach [13]. The TFISF idea works on the basis of information retrieval conclusion concept. Davakar Yadav and Vimal Kumar k [12] proposed the Hindi automatic summarized by the rule of extractive approach improvisation. They used the thematic based methodology which essentially depends on the frequently used words for the solution of Hindi. They divide the system architecture

Mr. Jazeb et al. [11] proposed the Urdu automatic summarized by the rule of extractive approach improvisation. They used the thematic based methodology which essentially depends on the frequently used words for the solution of Hindi. They divide the system architecture
into four steps. These four foremost steps are pre-processing, thematic base context, rank the sentences and generate the summary. Basically, they used pre-processing for the removal of Hindi stop words and remove the less or nothing meaning words by the analysis of Hindi morphology. After that, they used Thematic based method for the list of more frequently words by the help of word occurring in Input Hindi document. This step helps them in scoring process of sentence ranking and in last generates the Hindi text summarization. They used the gold standard summarize for the accuracy of automatic Hindi text summarization. They compare the system summary line by line with a human-generated summary. This Automatic Hindi text summarization achieves 85% by comparing with a human-generated summary. In future, they said that improvisation in this system for the multi-document Hindi automatic text summarization.

Mohammad Saleem Binwahlan [8] presented the Arabic text extractive summarization as the ESMAT. The author proposed the new method for the Arabic text document extractive summarization by the help of a combination of linear text features because linear features can produce more significant content from Arabic text documents for the automatic text summarization. The author uses the most features for the finding the relation of sentences among the document text. Proposed system fundamentally consists of two major steps. These two steps are pre-processing of Arabic text documents and extract the liners features for the output of autistic text summarization. This system first breaks down the input Arabic text document into sentences list, remove the common and stop words. In next step it does the extraction by the help of sentence concept, place of the sentence in the document and in the last step extracts the most conceptual sentence for the extractive Arabic summarization. The author compares this system generate summary with a human-generated summary of a text document with the help of ROUGE and achieve the good performance bench-marks as compare to the other Arabic automatic text summarizer.

Azadeh Zamaniifar and Omit Kashefi [11] proposed the AZOM Persian text summarize system. The authors build the text summarizer that fundamentally consider the conceptual meaning, structure of input text and look for a statistical approach for the summarization. They built the lexical backhand library for the determination of words relating to extracting the conceptual meaning of text document. This system consists of three major steps. These three steps include the pre-processing of Persian text document, extract the significant meaning of the text and produced the summary. In order to achieve the pre-processing of Persian text document, they used the segmentation for the boundaries of text sentences to remove the Persian stop words. Then they used lemmatizes process to extract the non-inflected form of text and make the whole text document as unified data and ready for the other processing. The interpretation step takes out the significant meaning of the whole document with the help of Persian language structure and the relation of words as a synonym, antonym etc. After that, statistical and conceptual weight is given to the input text sentences according to the most terms frequency, total frequency, lexical relation and corresponding terms. This proposed system also looks for the structural weight of sentences and blocks for the text summary. In last step of summary generation, those blocks of sentences that have the more weight according to the all above process were choose for the Persian text summarization. Aqil Burney et al. [5] proposed the Urdu summarize by the help of sentence weight algorithm. Basically, they built the Microsoft add-in for the Urdu text summarize. There are many algorithms for the text summarization but most of them are made for the European and English languages. Aqil Burney used the sentence weight algorithm for the Urdu automatic text summarization as an add-in for Microsoft. Sentence weight algorithm usually considered as a statistical approach. According to this algorithm every sentence of text document got the weight and sentences are ranked according to weight that will be used in output summarization. In the first step, it removes the stop words and extract the content words from input text file. The functionality of sentence algorithm is to calculate the total words in the text, then remove the stop words and calculate the content words in every sentence. Assign every sentence a weight according to the total content words. They used this algorithm for the Microsoft add-in. Basically, they introduce the initialize step for the Urdu text summarization and get some results by taking 20 documents of Urdu text. They achieve the 64% accuracy in the similarity of summarizing document by human and system generated a summary. They plan that in future more advanced technology for the elimination of stop words, content words extraction, scoring the sentence and ranked them will be used to achieve more accuracy in automatic Urdu text summarization.

III. URDU TEXT SUMMARIZER

Nowadays bulk volume of an Urdu data is growing so abundant in versatile behaviour like Urdu blogs, news, articles etc. As, Urdu is one of the leading speaking languages in sub-continent like Pakistan, India, and Bangladesh. So, there is also a necessity of Urdu text summarizer for the attainment of most imperative Urdu information from raw data. No accurate Urdu text summarizer exists nowadays. This paper aims the problem of Urdu text automatic summarization and suggested the resourceful system for the Urdu summarization. English languages communicate worldwide and catch attention in every arena of science. Unlike another language, Urdu did not catch the appropriate attention to researchers. This is the leading cause to encourage the authors to recommend the automatic Urdu text summarizer by the assistance of extractive summarization methodologies.

The authors got the idea of Urdu summarization for text documents. There is no proper or commercially available Urdu text summarization available for users. This thing makes difficult for us to choose the best algorithm for the Urdu language and collect the data for summarization. We choose the extractive base approach for our automatic Extract Urdu text summarizer (EUTS) system. There are many algorithms available for summarization but not favorable the Urdu summarization. There are some initial works for Urdu presented by different researchers and these works evidenced very handy for research work. We choose the improve weight algorithm for our approach and to achieve good results. The initial step of our proposed system architecture is pre-processing of Urdu text documents. This pre-processing step is furthermore divided into 5 sub-processes (see Fig. 1).
A. Sentence Segmentation

Segmentation is a text distribution process to limited units like words, phrases or sentences. The sentence segmentation applies to both humans and computer by mental practice in humans (when they read the text), and applied in the computer (when it comes to the field of natural language processing). Segmentation problems are unusual in Urdu because when someone clearly writes language with spaces, special characters in Urdu text, these special characters are not as same as other languages. The use of stop function / full stop in English and some other languages is reasonable for the sentence segmentation. It is quite a serious matter even in English because this problem is not too small due to the so much use of full stop character for the abbreviation which can delete any phrase/sentence or not. Along with the end of the word or sentence, all written languages include characters that are useful for the limits or boundary of the sentence. Sentence segmentation is useful for the parsing, information extraction and pre-processing of data.

After getting the input from user, first step is to make sections of the Urdu sentence from the document. Segmentation is basically used to clear out the boundary of sentences and this is a mandatory step in Natural language processing. Like other languages, Urdu has some values to show out the boundary of the Urdu sentences. This step proves itself very handy for our selected sentence weight algorithm.

B. Tokenization

All language pre-processing applications need input segmented text for more processing. The words are derived from the database on the base of the blank spaces and punctuation. The dataset also holds nonblank spaces or without punctuation words and variant non-Urdu words. To solve this problem, each word is seen from 'Tokenization' and it becomes a token. After getting the pre-process data of Urdu sentence from segmentation step, tokenization process is applied. Tokenization generally removes the extra spaces b/w the words and generate the list of Urdu words in the token form. For example:

```
احمد صدیق | نام | بابر | اسم | اسم | عبد | شیخ | اسم | اسم | اسم | اسم

Today in the country there are countless parties in the name of Muslim League.

[Today], [in], [country], [there], [are], [countless], [parties], [in], [the], [name], [of], [Muslim], [League].
```

C. Stop Words

According to Baeza-Yates and Rebiero-Neto, the more frequent words which act as a syntactic function in a language are stop words. They are often pointless and belong to closed classes. These lists are as following: The first list contains about 519 words [5]. The second list contains 500 words. This list is plotted by calculating frequency technique, in which nouns and named entities are not involved in huge Urdu corpus [7]. The third list contains 500 words [9]. The arranged stop word lists are used by some researchers, which should be obtained from domain corpus. The fourth list contains 195 words having closed classes. This list is plotted and used by open source resources of Urdu morphology. Words are separated on white-spaces in all stop words lists. From the above example of stopwords are:

```
[احمد], [صدیق], [بابر], [اسم], [اسم], [عبد], [شیخ], [اسم], [اسم], [اسم], [اسم]
```

D. Lemmatization

Urdu has very rich morphology due to the dependence of other languages of words. Urdu includes other languages like Arabic, Persian and Turkish etc. That’s why it is very difficult task for Urdu language pre-processing. Basically stemming and lemmatization are similar in a logical way but different in conceptual meaning. Stemming is a process to extract the word in root form by the emanation of suffix and prefix that includes spell checking word occurrence, word parsing and word counting etc. On the other hand, lemmatization removes the inflected form of the word by the help rich morphology or langue dictionary and show the
original root form of the word. Lemmatization is the linguistics approach to extract the word from grouping data or inflected form of word and process as the lemma word or original dictionary form of the word [6].

Lemmatization algorithm extracts the lemma word with the help of conceptual meaning in vocabulary apart from stemming. Lemmatization depends on the correct meaning of the word in a sentence and contextual meaning among the other sentences. The other main difference between stemming and lemmatization is that stemming operates on a word without the context knowledge while lemmatization is a very delicate process. We prefer the lemmatization upon stemming. Lemmatization is necessary because one word can appear in different plural or inflected form like

 COLORS (Colors), 책 (Books), كتاب (Books), كتاب (Book)

The above example shows that these are the same words but apparent in different inflected form. This can be easily judged by a human but the computer cannot easily detect that because it does not have a conceptual base dictionary. If we don’t apply the lemmatization on these words then the system would consider them as a single word that can reduce system accuracy. For other languages, lemmatization has mature rules and produce very good results but in Urdu lemmatization is not mature like other language that’s why we use our dictionary to use the lemmatization on Urdu. The basic condition for lemmatization process is that text should be in proper forms like proper sentences and the whole document should be in uniform text.

E. Sentence Weight Algorithm

The basic method we used for the sentence algorithm is word probability to find the word frequency in a sentence. Probability ‘P (w)’ finds the frequency of words ‘f (w)’ divides by the total no. of words ‘N’ after pre-processing as shown in the following equation.

\[ P(w) = \frac{f(w)}{N} \]

Equation 1

The importance of sentence is determined by the help of word probability equation. For each sentence weight, the average probability is equal to the importance of sentence in the document as shown in the following equation.

\[ g(S_j) = \frac{\sum_{W_i \in S_j} P(W_i) \times |W_i|}{\sum_{W_i \in S_j} |W_i|} \]

Equation 2

Where g(Sj) equals to the weight of sentence. The next step is to make the list of important sentences that contain the highest average probability of word. Apart from traditional sentence weight algorithm, we select the first sentence of topic or document as an important sentence and give him double weight with respect to other sentences. After ending the any paragraph the first sentence of next paragraph should get the 1.5 weight as compare to the other. In any textual document, there are many words that use many times but have no important effect on the main theme of the document; repeated words in the summary make it more weighted so we give them less weight as compare to singular words. Fig. 2 shows the proposed system working.

IV. RESULTS AND DISCUSSION

The accuracy and performance of the proposed system are then examined by the approach of the expert game. We asked a person to highlight or underline the most significant information of text documents. For the testing phase, we use the open source from Muhammad Humayoun and Hwanjo Yu on “github” a publish data set for the comparison of summary results [10]. These open source publish dataset is made up of different fields like news, health, current affairs, sports, and technology etc. This dataset also includes the human-generated summary. There are various procedures for evaluating the performance of system but we prefer precision, recall and F1 measure because these are the simple degree methods for evaluation.

Fig. 3 shows the results of sentence weight algorithm at the values of n from 1 to 5. Here we change the pre-processing setting. In the first step, we calculate the results by removing only stop words. After that, we calculate the results with own proposed system in which stop words and common words are excluded. When we calculate the results by removing only stop words than average re-call is 0.77185 and by excluding stop word and common words than average recall are 0.7826. When we calculate the results by removing only stop words then average precision is 0.4828766 and by excluding stop word and common words than average precision is 0.486273. When we calculate the results by removing only stop words then average F1 Measure is 0.5941328 and by excluding stop word and common words then average F1 Measure is 0.5930772.
Fig. 3. Results with IR Parameters

Fig. 4 shows the result of 50 documents separately. Here we see that every document has different Re-call, Precision and F1 Measure. On these documents we applied the sentence weight algorithm and check the results with a human generated abstracted summary at the value of n=1 in rouge. The overall system accuracy is 67% at the 50 documents as the comparison with human generated summary.

V. CONCLUSION AND FUTURE WORK

The basic method we used for the sentence algorithm is word probability to find the word frequency in a sentence. Probability finds the frequency of words divides by the total number of words after pre-processing. For each sentence weight, the average probability is equal to the importance of sentence in the document. The next step is to make the list of important sentences that contain the highest average probability of word. In last show the significant sentences as the user needed to display a number of sentences in the output of summarization.

The forthcoming awareness is to improvement from solitary article summary into the multi-document concise text system by disbursing this proposed method. For this improvement the mined summary has to be acquired from numerous documents those are connected with the perception. The expansion of this construction to several languages is furthermore a portion of the approaching option of struggle involved. In the future, we influence further procedures for the Urdu text summarization and associate the fallouts with existence methods. We famine to add extra pre-processing phases to rise the competence of the structure and mark him extra matured for Urdu text.

REFERENCES


