ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTOMATIC TEXT SUMMARIZATION OF INDIAN LANGUAGES: A MULTILINGUAL PROBLEM

A REVIEW OF MULTILINGUAL SUMMARIZATION TECHNIQUES

[1]JOVI D'SILVA, [2]Dr.UZZAL SHARMA

[1]Research Scholar, Assam Don Bosco University, Guwahati, Assam, India

[2]Assistant Professor, Assam Don Bosco University, Guwahati, Assam, India

E-mail: [1]jovidsilva@gmail.com, [2]druzzalsharma@gmail.com

ABSTRACT

Automatic text summarization is highly researched field. A lot of this research is limited to popular languages such as English. In a nation like India there are 22 languages spoken, which are written in 13 different scripts, with about 720 dialects. Taking this into consideration developing a nation-wide summarization tool for India would be a very difficult problem. In this paper we examine approaches to this problem and also highlight some existing research that has been done in Indian languages.

Keywords: *Automatic Text Summarization, Indian Languages, Multilingual, Language Independent*

1. INTRODUCTION

Summarization has been undertaken for as long as the written word has been around. Information and knowledge can be stored in concise manner using summarization.

The development of Text Summarization predates to the early 1950s, but with the increasing usage of the internet and web today, the technique of Automatic Text Summarization has gained importance. In addition to summarization by humans, we have machine summarization, where the machine creates the summary based on given inputs.

N. Verma and A. Tiwari mentioned in their paper that "A text extracted or generated, which is an important portion of an original text document(s), which conveys the information carried out by the original text(s), can be called a summary of that original text(s). When this is done automatically, i.e. with the help of a computer program, then we call this as automatic text summarization." [1].

With electronic media, summarization techniques have changed. Daily, we come across summaries that help us understand and process lots of information in a short time-span. News headlines, for instance, summarize the events of the previous day in just a phrase. Online, we find summaries of books, movies, articles. Even the minutes of a meeting is a summary, for it sums up the entire proceedings of a meeting in a few short and key points.

When summarizing, the person involved needs to have knowledge of the area in which the summarization needs to be done. This background knowledge makes for better and more accurate summaries. However, summaries are subjective as each summarizer may find different parts important and also have their own perspectives to their summaries. When summaries are machine-generated, there is no question of different perspectives or biases coming into the picture.

A mathematical model to understand human cognition does not exist. Therefore, human cognition is not computable. Since there is no way of pinpointing how a human summary has been come about exactly, machine-generated summaries cannot be fashioned based on the human model. If you give a machine a compression ratio, however, it will be able to decide for itself, the more important data of the input that needs to go in summary, from the less important data that can be omitted. Then, it will reformulate this important data to generate a summary.

Machine-generated summaries (or Automatic Text Summarization) come with its own difficulties and drawbacks. The machine cannot be fed with the same level of understanding and knowledge of the world as a human and it cannot be taught the language skills required when creating summaries of a given text. Thus, resorting to language tools becomes necessary. All of this makes Automatic Text Summarization a complicated process.

The World Wide Web has revolutionized the way in which knowledge is transmitted, consumed, and shared. Every year, the information that is made available on the internet, and the number of web pages, almost doubles. The number of documents that are available in languages other than English are also increasing rapidly. Besides consumers, the search engines too find it difficult to sieve through the huge volumes of information. Key-word based searches not only provide links to web-pages but also snippets of content in which these key-words have appeared. This provides the user with a mini-summary, like an indicative summary, before the decision of which web page to visit can be made.

Automatic Text Summarization seems to be the solution to deal with tons of information available online as well as offline [2] [3].

In a country like India, that has a vast lingual diversity, the problem of automatic text summarization becomes more grave if a common summarization tool needs to be developed for every language. The following sections examine multilingual summarization techniques which attempt to find a language independent solution to such a problem.

2. COMPARING TYPES OF SUMMARIES

Hovy et al. [4] have categorized the various types of summaries. Given below are popular categorizations of summaries produced in the literature.

2.1 Summarization Systems based on Input Documents

2.1.1 Single Document vs. Multi Document Summaries

These summaries are based on the type of input that is fed. Single Document summarisation is when the input consists of only one document. Multi Document Summarisation is when the input consists of a cluster of corresponding documents.

Multi Document Multilingual Summarisation takes place when the input is a cluster of multiple relevant documents in many different languages. A single language has to be chosen so that a summary can be generated in that language. Single Document and Multi Document summaries can be either extracts or abstracts [2] [3].

2.1.2 Monolingual vs. Multilingual Summarisation

Monolingual summarisation processes a single language, and produces the summary in that language. In contrast to this, we have multilingual summarisation.

Mani (2001) defines multilingual summarisation as “processing several languages, with summary in the same language as input.” [5] The summarisation tool works for a variety of languages. So if the input language is Konkani, the summary produced will be in Konkani and so on.

There is also a summarisation technique called Cross-Language Document Summarisation, in which, the summary is produced in a different language from that of the input or source document. For example, if the input is in English, the summary may be produced in Bengali.

Multilingual Summarisation is a more difficult than the other techniques because of the processing of documents in various languages, with different grammatical rules, and sentence syntaxes [2] [3].

2.1.3 Summarization Systems based on Genre

Summaries can be generated from a vast range of subjects. We can have summaries of news related input, technical scientific input, fictional works and so on.

2.1.4 Summarization Systems based on Length: Short vs. Long

The length of the summary generated depends on the length of the input. Source documents can sometimes be as short as one or two pages, and hence, their summaries will be shorter. Long input refers to source documents that are more than fifty pages long. Summaries generated for this type of input would to be proportionately longer.

2.2 Summarization Systems based on Purpose

2.2.1 Generic vs. Query Summaries

Summaries are intended for different users. Generic summaries are produced for those users who need summaries that work as surrogate documents so that they can avoid reading the source document. Precise summaries (query summaries) are produced for users who do not need the summary of an entire source document, but need specific details based on their query. The summaries are custom made to suit the various queries of the user, so only relevant data is utilised [3].

2.2.2 Indicative vs. Informative vs. Critical Summaries

A generated summary could be indicative, informative, or critical. An indicative summary indicates or merely points to the content in the input. It aids the user in the task of deciding whether the content is worth reading or not. On the other hand, an informative summary covers more details of the input document, like, its purpose, scope, results, and conclusion. Since the informative summary covers all the essential points of the input, it can be considered a surrogate document which can stand in place of the input as much as the reader need not read the principal document at all. However, whether this summary is entirely sufficient to satisfy the user's need depends on the depth of information required.

A critical summary aims to provide a critique of the document, rather than only providing information about its content. It contains the writer's views, opinions, or recommendations. A review is an example of a critical summary. The topic of the document is touched upon in brief, but pointing to the subject matter is not the objective of this kind of summary. This type of summary mainly evaluates the quality of the document i.e. strengths and weaknesses. Therefore, it could contain text that is not present in the source [3].

2.3 Summarization Systems based on Output

2.3.1 Extract vs. Abstract Summaries

An extract, is a type of summary that contains significant portions of the text provided in the input. An abstract, which is another type of summary, embodies the subject matter of the input (alternatively referred to as a source or a principal document). It represents the input with text units formed by redrafting the noteworthy portions chosen from the input. Both extracts and abstracts are based on input, and the summary that is created by the summariser.

3. SUMMARIZATION SYSTEMS BEING DEVELOPED IN OTHER INDIAN LANGUAGES

The following methods have already been used to work on Indian languages, to create summaries:

Hindi: Manjula Subramaniam and Vipul Dalal (2015) have used the abstractive method using rich semantic graph techniques. [6] Dr Latesh Malik (2013) has used the extractive method for single document summarization. This method has used statistical and linguistic features, as well as Genetic Algorithm. [7] The extractive method has also been used by Anita J. et al (2014) using features, Fuzzy Classifier and Neutral Network. [8]

Punjabi: Vishal Gupta carried out Text Summarization in the Punjabi language in 2010 and 2012. The extractive method, for both, the news document as well as the text has been used. News document summarization consisted of two phases: pre-processing and processing, while for text summarization, the TF-IDF technique was used. [9]

Tamil: Banu et al (2007) [10] and Kumar and Devi (2011) [11] have all carried out work in Tamil summarization. All three used the extractive method for text summarization. Banu used the semantic graph. Kumar and Devi used graph theoretic scoring technique.

Bengali: In Bengali language, Islam and Masum formulated Corpus Oriented Text Summarization System called 'Bhasa'[12]. Sarkar used the extractive method as well, utilizing TF-IDF techniques [13]. Das and Bandopadhyay (2010) who utilized the extractive method, used the k-means technique and the page rank standard methodology [14].

Kannada: Kallimani et al have put forth a text summarizer for the language Kannada called "AutoSum" [15]. Kannada texts were worked on in 2011 and 2012 by Jayashree [16], and Jayashree R et al [17], respectively. Both times the extractive method was used. Whereas, in 2011, TF-IDF techniques and in 2012 GSS coefficient TF-IDF techniques were used.

Malayalam: Ajmal E.B. and Rosna P. Haroon (2015) used the extractive method for text summarization in Malyalam[18]. Maximum Marginal Relevance (MMR) techniques were used with successful threshold. In 2014, Kabeer and Idicule used the extractive method using Navie Bayes, Neutral Network, and the HMM model of machine learningtechniques [19]. In 2015, Renjith et al used the extractive method using TF-IDF techniques [20].

Telegu: Telegu texts were worked on using abstractive methods, two times. Vekateshwar (2011) used ontology approaches [21], whereas Kallimani et al used IE rules, and class-based templates [22].

In addition, Rosna P Haroon worked on text summarization methods in Dravidian language contains languages like Malayalam, Tamil, Telugu, Kannada, Kodagu, Badaga, Byari, and Tulu. [23].

Odia: Odia texts were summarized after stemming by R. C. Balabantaray et al. (2012) They calculated scores for individual words in the documents and similarly assigned sentence score by adding the scores of all the words in the sentence and then dividing the value by the total word count in that sentence. The summarizer then extracts top ranking sentences and also the first sentence of the first paragraph for inclusion in the summary. [24].

Marathi: Yogeshwari V. Rathod (2018) performed text summarization on Marathi news articles using a Graph Theoretic approach using PageRank algorithm. [25].

4. MULTILINGUAL APPROACHES APPLIED TO INDIAN LANGUAGES

Sarkar and Bandyopadhyay were the pioneers of multilingual text summarization [24]. They put forward a design for multilingual summarization for Indian languages. This was a news summarizer which received news streams from various online newspapers in different languages. These were then directed into numerous output news streams by using events. News streams in different languages of the same ev ent were matched and combined into a cluster. The most representative sentences in each cluster were then used to generate the summary.

Patel et al developed a proficient algorithm for language independent generic extractive summarization for single documents [25]. The algorithm is in accordance with structural and statistical components rather than semantic factors. Single-document summarization for languages like English, Hindi, Gujarati and Urdu was evaluated.

Perumal et al (2011) put forth a language independent procedure for single documents automated summarization hinged on sentence extraction [26]. The proposition employed structural characteristics based on sentence scoring along with PageRank that relies on ranking sentences. This approach works for English and Tamil documents.

Keyan employed neural networks for presenting multi-lingual (Tamil and English) multi-document summarization [27]. The system entails three primary steps. In step one; the sentences are changed to vector form. The next step involves values of weights that are allotted to vector form with reference to sentence characteristics. Then, single document summarization is performed depending on sentence weight value. The output of single document summarization is taken as an intake for multi-document summarization. The third and final step involves selection of sentences, where output summary is picked based on similarity and dissimilarity scales which are utilised to compare the sentences.

Gupta (2014) [28], examines hybrid text summarization algorithm independent of language. This text summarizer uses seven features: (1) Words that are similar to the title line (2) n-gram similarity with title (3) Normalized NTF-PSF feature (4) Position attribute (5) Relative length (6) Numerical data extraction (7) Domain specific keywords features specified by the user. All the language independent features rest on statistics. No feature specific to language is considered, except a list of stop words and stemmer for that particular language. In this, language independent summarizer, equal importance is given to every feature; hence no weights have been allocated to various features. The top scored 20% sentences are drawn out and re-positioned in accordance with their emergence in the intake. This is done to maintain sentence coherence.

5. TOWARDS BUILDING LANGUAGE INDEPENDENT SUMMARIZATION SYSTEMS

In India, the text summarization problem will vary across geographical boundaries because of the numerous languages spoken on the sub-continent. An approach used for summarizing Malayalam, may not be suitable for Punjabi or Bengali. Varying scripts, syntax of sentences and grammatical structures make it necessary to use different approaches for the same job of text summarization.

A language independent outlook to text summarization would involve developing an algorithm that can work across languages no matter what script, or syntax the language uses. The algorithm would have to create a summary of the input irrespective of the language of the input.

Most of the previous attempts at developing summarization tools have been limited to a single language and these tools could not be extended to other languages. However, work done by each of [24], [25], [26], [27] and [28], attempts at developing multilingual systems that work with the respective language and also can be extended to work with other languages. Yet, there are numerous language independent methods that need to be examined with respect to Indian languages and this paper provides a detailed review of the popular methods in the field.

6. LANGUAGE INDEPENDENT MULTILINGUAL APPROACHES

6.1 Language Independent Multilingual Single-document Summarization Approaches

6.1.1 Position Based Methods

According to Lin et al. [29] Position Method builds on the observation that the discourse structure of texts of a particular style are usually predictable, and that sentences of higher topic centrality are more likely to appear in certain definable positions. Therefore, it is the positional data of a sentence in a document that is utilized to estimate the score for a given sentence. A popular method is Lead Based Method, in which a sentence that has already occurred in a given document has more significance. The score of the sentence is measured as shown below,

$$\text{Score}(\text{Si})=\frac{1}{\text{i}} \qquad (1)$$

Some others that use positional based sentence scores allot higher scores to sentences that appear at the beginning or the end of paragraphs [30] [3].

6.1.2 Edmundson's Methods

In 1969, Edmundson[31] carried out experiments with a dataset that consisted of 200 scientific articles of chemistry. The following are the methods that were used:

Cue Words: Cue Words Method uses cue words to give a score to a sentence. The cue weight of each sentence is the summation of cue weights of the words that appear in a sentence.

Title Words: Title Words Method determines a score of a sentence in accordance with the sum of title weights of the words occurring in a sentence.

Key Words: In this technique, a score is attributed to the sentence depending on the total sum of the weights of keywords in a sentence.

Sentence Position: The Sentence Position Technique allots a score to a sentence based on its location in the document.

The initial three approaches mentioned above assign scores to sentences that are derived from word level features after pre-processing of the word in the document. The fourth method considers the position of a given sentence. These features are then linearly combined with the help of the following formula,

$$\text{Score(S)}= \text{a1}\times\text{C} +\text{a2}\times\text{K} +\text{a3}\times\text{T} +\text{a4}\times\text{P} \qquad (2)$$

Where C is Cue word, K is Key word, T is Title and P is Sentence Position and a1, a2, a3, and a4 are feature weights for the four features respectively. These four feature weights are the ones that are manually tuned by cross referencing documents and the manually composed extracts. Edmundson's assessments on test data show that Key Words Method is weaker than the other three features. The blend of ‘cue-title-location’ was the finest. As an individual feature, ‘Location’ is excellent. ‘Key Words’ is poor as an individual feature. Researchers building on Edmundson’s work have used variations of these features [30] [3].

6.1.3 LUHN’s Method

Luhn [32] brought out the idea that a document is made up of some words that describe the matter in the document. The sentences present in the document which express the most important information are the ones with the maximum of such descriptive words very near to each other [33]. He also recommended using occurrence frequency to establish which words explain the topic of the document. He fashioned a predetermined list known as ‘stop word list’, consisting of words alike, to eliminate them from review. Another category of words that do not occur in the stop word list, but cannot still be indicative of the subject of a document was developed. Luhn used empirically determined high and low frequency thresholds. The high thresholds screened out words which arose too often throughout the article. The low thresholds screened out words that arose very infrequently. The words that remain are descriptive words and indicate the important content. Therefore, the sentence score computed is dependent on the count of descriptive words within the sentence and the linear distance between them because of the interference of non-significant words. Luhn recommends that the ideal limit of the intervening trivial words between the bracketing descriptive words should be set to 4 or 5 words that are not significant.

The Score of the ith portion of the sentence bracketed by descriptive words is calculated using the following formula,

$$\text{Score}_i = S_i^2\, N_i \qquad (3)$$

Where S_i is the total count of descriptive words in the ith portion and N_i is the total word count in the ith portion. Following the score computations for the various sections of the sentence S, the total score for the sentence is calculated as follows,

$$\text{Score(S)} = \max_{i \leq C}\{\ \text{Score}_i\} \qquad (4)$$

Where, C is the part of the sentence S bracketed by keywords. Score_i is the score of the ith portion of a sentence [30] [3].

6.1.4 TF*ISF and TF*IDF Methods

Neto et al. [34] suggested an Extractive Text Algorithm premised on the significance of the areas present in a document. In this style, the document is first divided with the help of the Text Tiling Algorithm that recognises topics (Consistent text segments) based on the TF-IDF (Term Frequency - Inverse *Sentence* Frequency) metric. Then, for individual topics, the algorithm estimates a measure of its relative significance in the document. This measure is computed by using the notion of the TF-ISF (Term Frequency - Inverse Sentence Frequency) metric. Finally, the summary is formed by choosing several sentences from each area proportional to the importance of that area. Just like TF-IDF, TF-ISF stands for the product of term frequency and inverse sentence frequency. The score for a textual unit is computed as an average TFISF for all words in a textual unit [30][3],

$$\text{Score(S)}= \sum_{t \in S} f(t) \text{x}\ isf(t) \qquad (5)$$

Where, *f(t)* is the frequency of occurrence of the term and *isf(t)* is computed as follows,

$$\text{isf(t)}=1-\frac{\log(nt)}{\log(n)} \qquad (6)$$

Where, *n* is the total count of the sentences in the document and *nt* is the total count of the sentences that contain *t*.

6.1.5 SVD Methods

Singular Value Decomposition [35] is a vital section of LSA (Latent Semantic Analysis) that is entirely an automatic algebraic-statistical technique for extracting and representing the contextual management of word meanings in passages of discourse. According to [36], one should construct a term by sentences m × n matrix A = [A1, A2,...,An], where each column A_i depicts the weighted term-frequency vector of the ith sentence in the document, and apply SVD to the matrix A: $A = U\Sigma V^T$. The summarization method proposed by [8] chooses the sentences for the summary based on the relative magnitude of the topics they state, described by the matrix V^T. The algorithm for summarization merely chooses the most crucial sentence for each topic: i.e., the kth sentence chosen is the one with the maximum index value in the kth right singular vector in matrix V^T. The Enhanced Summarization Method initiated by [35], picks the sentences whose vectorial representation in the matrix $\Sigma^2 \cdot V^T$ has the highest 'length'. Intuitively, the plan is to select the sentences with the greatest collective weight across all significant topics. In more formal terms, sentence score is computed as a length of the sentence vector in $\Sigma^2 \cdot V^T$ after computing SVD [30] [3].

6.1.6 Graph Based Methods

R. Mihalcea presented a multilingual version of TextRank [37] not including morphological analysis. Each document is depicted as a nodes graph that stands for sentences interlinked by similarity (overlap) relationship. The intersection of two sentences is established in the simplest way as the number of similar tokens between the two sentences, normalized by the length of these sentences. In other words, given two sentences S_i and $S_{j,}$ where the sentence S_i is depicted by the set of N_i words: $w_1, w_2, \cdots, w_{Ni}$, the similarity of S_i and S_j is defined as,

$$\text{Similarity}(Si,Sj)=\frac{|\{\omega k|\omega k\in Si\&\omega k\in Sj\}|}{\log(|Si|)+\log(|Sj|)} \qquad (7)$$

The sentence score is equal to the PageRank [38] score of its node in the representation graph,

$WS(V_i)=$

$$(1-d)+d\times\sum_{Vj\in In(Vi)}\frac{\omega ji}{\sum Vk\in Out(Vj)\omega jk}WS(Vj) \qquad (8)$$

Where, $ln(V_i)$ is the set of vertices that point to V_i (predecessors), $Out(V_j)$ is the collection of vertices that it points to (successors) by vertex V_j, d is the damping factor which merges into the model the probability of jumping from a given vertex to another arbitrary vertex in the graph (value used for d = 0.85, setting the probability of jumping to a untried node at 0.15), and w_{ji} is the weight designated to the edge linking the two vertices: V_j and V_i and equivalent to the similarity value amid the corresponding sentences [30] [3].

6.1.7 Genetic Algorithm Methods

Last et al. [30] put forth an original technique called "MUSE (Multilingual Sentence Extractor)" to "language-independent" extractive summarization. This approach depicts the summary as an aggregation of the highly elucidative fractions of the summarized text with no language-specific text analysis. A Genetic Algorithm was implemented to discover the finest linear combination of 31 sentence scoring indicators based on vector and graph representations of text documents. Here, summarization is reasoned as an optimization or a search problem. In accordance with this methodology, an aggregation of features is linearly combined to score sentences in a document. The genetic algorithm is then employed to find the superior weight configuration utilized for feature combination. The weight model studied for one language is referred to another language to establish proficiency of the model across various languages [3].

6.1.8 Statistical Approach

Patel et al [25] brought forward a language independent generic extractive summarization for single documents which use structural and statistical factors (rather than semantic factors) to make the process language independent. The technique bases its foundation on the fact that diverse languages involve diverse complexities of their own semantics, causing it extremely difficult to employ natural language processing. Whereas, a statistical methodology is quite robust and can effortlessly be adapted to various languages. Nonetheless, the approach requires a stop words list (supplied externally) and a stemmer for equivalent languages in which documents are to be summarized [3].

7. LANGUAGE INDEPENDENT MULTILINGUAL MULTI-DOCUMENT SUMMARIZATION APPROACHES

There are numerous multi-document text summarization techniques for English[3]. The existent multi-document Summarization methods with properties of language independence can be effortlessly extended to multilingual summarization, but there are only a restricted number of similar techniques that have been accurately verified on multilingual datasets. Language independent single document text summarization features such as TF-IDF, Position, Title/Headline and Centroid have been broadened for use in language independent multi-document multilingual summarization tasks [3].

MEAD [39, 40] is a text summarization platform that makes use of features, classifiers and re-rankers to identify what sentences are to be included in the final summary. The default features that are included are centroid, position and sentence length. These are compounded into a composite score for every sentence, though there are other features implemented.

The centroid value of each word that is present in a sentence is estimated by multiplying the term frequency (tf) of a given word by that word's inverse document frequency (idf) obtained from a corpus. The 'tf' of a word is computed by dividing the number of times the word occurs in a document cluster by the total count of terms in the document. The 'idf' is calculated by dividing the total number of documents by the total count of documents with the word in it and finally taking the log value of the resulting division.

The centroid value of a sentence is the aggregate of the centroid values of the words in the sentence,

$$Ci = \sum_{\omega} C\omega, i \qquad (9)$$

The positional value Pi of a sentence is calculated using the formula,

$$P_{Si=} \frac{\boldsymbol{n-i+1}}{\boldsymbol{n}} x\mathit{Cmax} \qquad (10)$$

Where, n illustrates the number of sentences in the document, 'i' illustrates the position of the sentence inside the text, and $C_{max}$is the record of the sentence that has the maximal centroid value.

The length feature is used as a cutoff feature such that any sentence with a length shorter than the threshold (tresh) will receive a score of zero, irrespective of other features. The sentence which has a length greater than the specified default 'tresh' will receive a score that is an aggregation of other features, as there is no algorithm to predict the weight of the features used we use equal weight for all the features (W_c and W_p is equal to one), this is shown in the following equations,

$$Score\ (S_i)=W_cC_i + W_p\,P_i\ ;\text{if Length } (S_i) > \text{tresh} \qquad (11)$$

$$Score\ (S_i) = 0;\ \text{if Length } (S_i) < \text{tresh} \qquad (12)$$

According to D. Radevet al. [40] "Four classifiers come with MEAD". Default: offers a linear combination of all features excluding 'Length' which is handled as a cutoff feature, Lead-based: a baseline classifier that favours sentences that arise sooner in the cluster, as specified by the order of documents in the definition of the cluster, Random: a baseline classifier that extracts sentences at random from the cluster and "Decision-tree: a machine learning algorithm", based on Weka (Witten and Frank, 2000) and trained on "an annotated summary corpus". Lastly, the re-rankers are then used to amend the scores of a sentence founded on their relation with other sentences.

Another summarizer NewGist by Kabadjov et al [41] is a multilingual statistical news summarizer that utilizes a language independent method to multi-document text summarization. This technique uses "Singular Value Decomposition (SVD)" for noteworthy sentence selection. Since SVD has the edge of being language independent, the procedure has been applied on diverse languages.

8. MULTILINGUAL TEXT SUMMARIZATION AS MACHINE TRANSLATION TASK

Machine translation (sometimes abbreviated to MT), is a sub-field of computational linguistics. It examines the utility of software to transform text or speech from one language to another. Machine Translation is thus a method of allowing an existing summarizing system to manage multiple languages.

One of the ways of solving the problem is by a machine translating the output (generated summaries). Another technique is by machine translating the input documents to the required language of output, before the sentence extraction phase. The first means of dealing with this issue is preferable, as translating the input in advance makes the sentence weighting uncertain. A machine translating the input could prove to be problematic, as if errors occur during this process; these errors will persist in the summary generated [42].

Examples of the above: A summary system like "SUMMARIST" [43] extracts summaries from sentences in a range of languages and then translates the summary that emerges. On the other hand, a system like "NewsBlaster" translates the document before extracting the sentences.

The output produced by machine translating systems usually contains inaccuracies that make the summary even less readable. This becomes an even bigger problem for languages that are linguistically distant: e.g. English, Mandarin, and Russian [42].

9. CHALLENGES IN DEVELOPING MULTILINGUAL SYSTEMS

Developing multilingual summarisation systems can pose certain challenges. We need to consider these challenges, and make them known when we work with multiple languages. Here are some of the factors we need to keep in mind:

9.1 Tokenization

Each language is coded differently. The way English is coded is not the same as the way Arabic is coded. Tokenising is relatively simple when summarising the English language. Token boundaries are easily identifiable because of the white-space around words, as well as punctuation that indicates the end of a sentence. However, all languages cannot extract tokens in the same way. In Cantonese or Mandarin for example, tokens need to be extracted from text that contain no white-spaces as indicators. Summarization systems that depend on sentences, and not words, punctuation can create a problem. In the Roman script, a full-stop (.) generally indicates the end of a sentence. However, abbreviations also use the same indicator (.), but this does not indicate the end of a sentence. For e.g. “When Dr. Alvarez came over, we were out.” The sentence does not end at the use of the first (.) after “Dr.”

9.2 Anaphoric Expression

In monolingual summarisation, anaphora (pronouns, definite noun phrases, discourse markers) can be recognised in order to make a more structured summary. In multi-lingual summarisation, anaphora cannot be identified in the same way as names could appear differently, and discourse markers could have different semantics.

9.3 Discourse Structure

Every language has its own structure, and understanding the structure of a language helps to make better summaries. Different languages, however, may utilise an entirely different structure to convey the same text/meaning of a text.

9.4 Machine Translation

Machine translation technology has not yet reached the state of perfection, and so, quality cannot be ascertained. Therefore, while designing multilingual summarisation systems that utilize machine translation, developers need to be aware when it is feasible to use machine translation and when it is not. This also depends on whether summarisation is done before or after identifying tokenizers.

Each of the above mentioned challenges are independent research areas of natural language processing. Further research and progress in these fields would aid in the advancement in the field of automatic text summarization and thus multilingual text summarization.

10. EVALUATIONS

In order to grade the quality of a system-generated summary, we require the gold-standard summary of the input document, created by the human. The system-generated summary can then be contrasted and compared with the human summary, and its quality can then be determined based on this. This is typically done with the help of ROUGE Toolkit. The toolkit recommends a minimum of two human generated summaries, with which to compare the system-generated summary.

11. CONCLUSIONS

A large number of the prior attempts at developing summarization tools for Indian languages have been limited to a single language and could not be extended to other languages. There has been some work done in developing multilingual systems that work with a particular language and can also be extended to work with other languages. There are numerous language independent methods that were yet to be examined with respect to Indian languages.

In this paper, various Language Independent Multilingual approaches have been examined as potential solutions towards developing a nation-wide summarization tool for India along with possible challenges that could be encountered in the process.

Approaches towards text summarization in a country like India can be particularly challenging as there are many languages spoken across the country, each having different dialects, scripts and grammatical structures.

In this scenario, a language independent approach for text summarization can prove to be enormously constructive as the algorithm would have the potential to create summaries irrespective of the language of the input text.

REFERENCES

[1] N. Verma, A. Tiwari, “A Survey of Automatic Text Summarization”, *International Journal of Engineering Research & Technology (IJERT)*, Vol. 3 Issue 6, Jun. 2014, pp.258–263.

[2] Hmida Firas, "Language Independent Summarization Approaches”, *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*, IGI Global, 2014, pp. 295-307. Web. 2 Sep. 2017. doi:10.4018/978-1-4666-5019-0.ch013.S

[3] Kamal Sarkar, "Multilingual Summarization Approaches", *Computational Linguistics: Concepts, Methodologies, Tools, and Applications*, January 2014, pp. 158.

[4] Eduard Hovy, Chin-Yew Lin, "Automated Text Summarization and the SUMMARIST System." *Advances in Automatic Text Summarization*,Baltimore, Maryland,Association for Computational Linguistics, October 1999,pp. 13-15.

[5] Inderjeet Mani, Mark T Maybury, “Automatic Summarization”, *John Benjamins Publishing Company* 3, Vol.3, 2001.

[6] Manjula Subramaniam, Vipul Dalal, “Test Model for Rich Semantic Graph Representation for Hindi Text using Abstractive Method”, *International Research Journal of Engineering and Technology (IRJET),*Vol.2(2), 2015.

[7] Chetana Thaokar, Latesh Malik, "Test Model for Summarizing Hindi Text Using Extraction Method", *IEEE Conference on Information & Communication Technologies (ICT)*, 2013, pp. 1138-1143.

[8] J Anitha, PVGD Prasad Reddy, M S Prasad Babu, “An Approach for summarizing Hindi Text through a Hybrid Fuzzy Neural Network Algorithm”, *Journal of Information and Knowledge Management*, Vol. 13, No. 4, 1450036(2014).

[9] Vishal Gupta, Gurpreet Lehal, “Complete Pre-Processing Phase of Punjabi Text Extractive Summarization System”,*Proceedings of COLING 2012: Demonstration Papers,*COLING 2012, Mumbai, India, pp. 199–206.

[10] M. Banu, C. Karthika, P Sudarmani, T.V. Geetha, “Tamil Document Summarization Using Semantic Graph Method”, *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) 2*, 2007, pp. 128-134.

[11] S. Kumar, V. S. Ram, S. L. Devi, “Text Extraction for an Agglutinative Language”, *Proceedings of Journal: Language in India*, 2011, pp. 56-59.

[12] Md. Tawhidul Islam, Shaikh Mostafa Al Masum, “Bhasa: A Corpus Based Information Retrieval and Summarizer for Bengali Text”, *Proceedings of the 7th International Conference on Computer and Information Technology*, Macquarie University, Sydney, Australia, 2004.

[13] Kamal Sarkar, “Bengali text summarization by sentence extraction”, *Proceedings of International Conference on Business and Information Management(ICBIM-2012)*, NIT Durgapur, India, 2012, pp. 233-245.

[14] Amitava Das, Sivaji Bandyopadhyay, “Topic-Based Bengali Opinion Summarization”, *23rd International Conference Computing Linguistics: Posters*, Beijing, 2010, pp. 232–240.

[15] Jagdish S. Kallimani, K.G. Srinivasa, “Information Retrieval by Text Summarization for an Indian Regional Language”, *Proceedings of 6thInternational Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010)*, 2010, pp. 1-4.

[16] R. Jayashree, K. M. Srikanta, K. Sunny, “Document Summarization in Kannada using Keyword Extraction”, *Proceedings of AIAA 2011,CS& IT 03*,2011, pp. 121–127.

[17] R. Jayashree, “Categorized Text Document Summarization in the Kannada Language by Sentence Ranking”, *Proceedings of 12th International Conference onIntelligent Systems Design and Applications (ISDA)*, 2012, pp. 776-78.

[18] E. B Ajmal, Rosna P Haroon, “Summarization of Malayalam Document Using Relevance of Sentence”, *International Journal of Least Research in Engineering and Technology (IJLRET),*Vol. 1, Issue 6, 2015.

[19] Rajina Kabeer, Sumam Mary Idicule, “Text Summarization for Malayalam Documents – An Experience”, *International Conference of*

Data Science and Engineering (ICDSE), 2014, pp.145-150.

[20] S.R Renjith, P Sony, “An Automatic Text Summarization for Malayalam Using Sentence Extraction”, *Proceeding of 27th IRF International Conference,* 2015.

[21] K Venkateshwar Rao, “New Directions in Automated Text Summarization”, *Jawaharlal Nehru Technological University*, 2011.

[22] Jagadish S Kallimani, K G Srinivasa, B. Eswara Reddy, “Statistical and Analytical Study of Guided Abstractive Text Summarization”,*Current Science*, Vol. 110, No. 1,2016, pp.69.

[23] Rosna P Harun, “Text Summarization methods in Dravidian Language”, *International Journal of Innovations in Engineering and Technology (IJIET)*, Vol. 7, Issue 1, June 2016.

[24] Kamal Sarkar, Sivaji Bandyopadhyay, "A multilingual text summarization system for Indian languages", *Proceedings of Second Symposium on Indian Morphology, Phonology and Language Engineering: Simple 2005*, IIT Kharagpur,, India, February 5th-7th, 2005.

[25] Patel Alkesh, Tanveer Siddiqui, Uma Shanker Tiwary, "A Language Independent Approach to Multilingual Text Summarization", *Large scale Semantic Access to Content (text, image, video, and sound)*, Pittsburgh, Pennsylvania, May 30 - June 01, 2007, pp.123-132.

[26] Krish Perumal, Bidyut Baran Chaudhuri, "Language independent sentence extraction based text summarization", *Proceedings of ICON-2011: 9th International Conference on Natural Language Processing*, Macmillan Publishers, India, 2011.

[27] M. Karthi Keyan, K.G. Srinivasagan, “Multi-Document and Multi-Lingual Summarization using Neural Networks”, *International Conference on Recent Trendsin Computational Methods, Communication and Controls (ICON3C 2012), Proceedings published in International Journal of Computer Applications (IJCA)*, 2012, pp. 11-14.

[28] Vishal Gupta,"A Language Independent Hybrid Approach for Text Summarization", *Emerging Trends in Computing and Communication. Springer*, New Delhi, 2014, pp. 71-77.

[29] Chin-Yew Lin, Eduard Hovy, "Identifying Topics by Position", *Proceedings of the fifth Conference on Applied Natural Language Processing*, Association for Computational Linguistics, 1997.

[30] Mark Last, Marina Litvak, "Language-independent Techniques for Automated Text Summarization", *NATO Science for Peace and Security Series - D: Information and Communication Security*, Ebook Vol. 27: Web Intelligence and Security,2010, pp. 207-237.

[31] Harold P Edmundson, "New methods in Automatic Extracting",*Journal of the ACM (JACM),* Vol. 16, Issue 2, April 1969, pp. 264-285.

[32] Hans Peter Luhn, "The Automatic Creation of Literature Abstracts",*IBM Journal of Research and Development,*Vol. 2, Issue 2, April 1958, pp.159 - 165.

[33] Ani Nenkova, Kathleen McKeown, "Automatic Summarization", *Foundations and Trends in Information Retrieval (2011)*, Vol. 5. No. 2–3, 30 June 2011, pp. 103-233.

[34] Joel Larocca Neto et al.,"Generating Text Summaries Through the Relative Importance of Topics", *In: Monard M.C., Sichman J.S. (eds) Advances in Artificial Intelligence. IBERAMIA 2000, SBIA 2000*. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 1952, pp 300-309.

[35] Josef Steinberger, Karel Jezek, “Text Summarization and Singular Value Decomposition”, *In: Yakhno T. (eds) Advances in Information Systems. ADVIS 2004*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Vol. 3261, 2004, pp. 245–254.

[36] Yihong Gong, Xin Liu, “Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis*”, In: Proceedings of the 24th ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 19–25.

[37] Rada Mihalcea, “Language Independent Extractive Summarization”,*In: AAAI’05: Proceedings of the 20th national conference on Artificial intelligence*, 2005, pp. 1688–1689.

[38] Sergey Brin, Lawrence Page, “The Anatomy of a Large-Scale Hyper-textual Web Search Engine”, *Proceedings of the Seventh International World Wide Web Conference*,Computer networks and ISDN systems, Vol.30, Issues 1-7,April 1998, pp.107–117.

[39] Rada Mihalcea, Hakan Ceylan, “Explorations in Automatic Book Summarization,” *Proceedings of the Joint Conference on Empirical Methods in Natural Language*

Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, June 2007, pp. 380–389.

[40] D. Radev et al., “MEAD - A platform for Multi Document Multilingual Text Summarization”,*International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, May 2004, pp. 699–702.

[41] Mijail Kabadjov et al. "News Gist: A multilingual statistical news summarizer",*Machine learning and knowledge discovery in databases,*ECML PKDD 2010, Springer, Berlin, Heidelberg, pp. 591-594.

[42] Torres-Moreno, Juan-Manuel, ed. “Automatic text summarization”, John Wiley & Sons, 2014.

[43] David Kirk Evans, Judith L. Klavans, Kathleen R. McKeown, "Columbia Newsblaster: Multilingual News Summarization on the Web", *Demonstration Papers at HLT-NAACL 2004*, Association for Computational Linguistics, 2004.