IMPACT FACTOR 5.473(SJIF)
UPA NATIONAL E-JOURNAL
Interdisciplinary Peer-Reviewed Indexed Journal
Volume -8 : Issue-1 (February-2022)
ISSN 2455-4375

SUMMARIZATION OF AUTOMATIC TEXT SIMPLIFICATION USING NATURAL LANGUAGE PROCESSING

Dr. Pankaj V. Nimbalkar
Assistant Professor,
Department of Computer Science
Dr. Ambedkar College,
Deekshabhoomi, Nagpur-10
Email:pankajnimbalkar3@gmail.com
Mobile No:9011095977

Abstract : *The world of internet is getting* ***exploded*** *with a bulk amount of data every day, being able to automatically summarize is big challenge. Summaries of long documents articles in news, or even conversations can help us consume content faster and more efficiently. Automatic Text Summarization is a growing field in NLP and has been getting more attention in the last few years.*

Keywords: NLP,Text simplification.

---

1. Types of Text Summarization

Two types of text summarization methods are extractive and abstractive. **Extractive summarization** is necessarily picking out sentences from the text that can best represent its summary. Extractive summarization techniques have been common for quite some time now, owing to its origin in 1951. It's more about learning to understand the importance of each sentence and their relations with each other rather than trying to understand the content of the text.

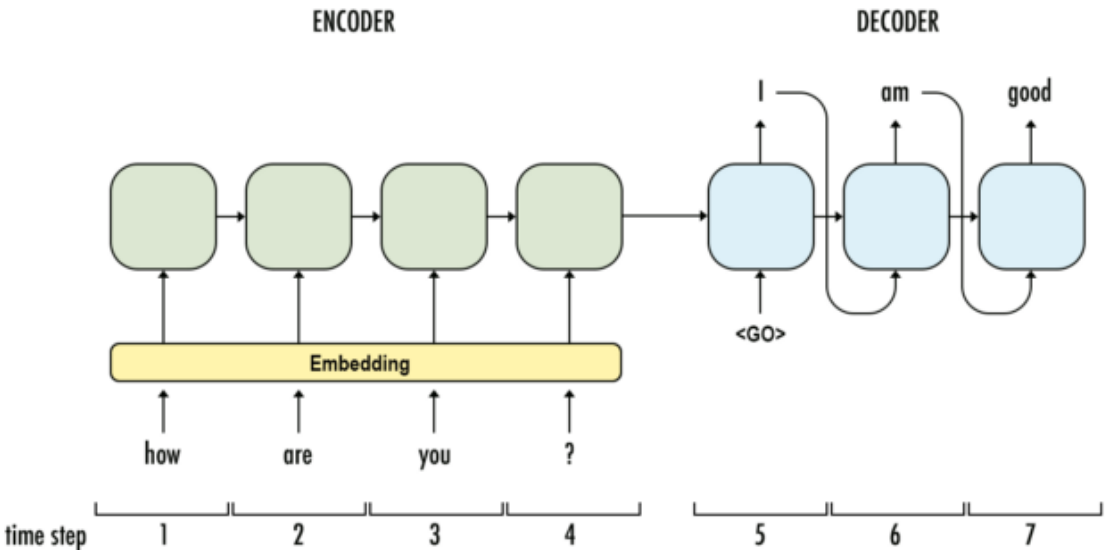
1.1 Abstractive summarization, on the other hand, is all about trying to understand the content of the text and then providing a summary based on that, which may or may not have the same sentences as present in the original text. Abstractive summarization tries to create its own sentences and is definitely a step towards more human-like summaries.

The techniques working to do extractive and abstractive summarization are different from each other. Extractive summarization is, crudely speaking, a sentence ranking problem while abstractive summarization involves more complex linguistic models as it generates new sentences.

1.2 Encoding and decoding Techniques:

since the arrival of Deep Learning and Abstractive Summarization, Interaction with machines through natural language and Machine Translation have all been getting a lot of success. Due to parallelism *Machine Translation and interaction* follows with Abstractive

Summarization. All of these techniques **encode** an input sentence into features and then tries to generate a different sentence i.e. **decode** these features.



A commonly used Deep Learning based Machine Translation model is an LSTM based Encoder Decoder network with Attention. The model starts with an LSTM based Encoder which converts the sentence into a vector of features. The decoder, also made up of an LSTM, is responsible for creating the output, one word at a time. The decoder starts with the vector of features provided by the encoder and then each word is predicted based on the previous word prediction and LSTM output. Attention is placed on the encoder features to make them even more specific to the current word.

Creation of new sentences is a complex process that the machines have not mastered yet. An issue with Abstractive Summarization is also the length of sentences to be encoded. While LSTMs have the ability to capture both long term and short term contexts, even they have a limit for long term. This makes summarizing really long documents difficult.

Another astronomically important issue for summaries is that it should never contain facts that contradict the input text. Extractive summarization can never face this problem since they pick up sentences directly from the text. But abstractive summarization is prone to such factual incoherence.

1.3 Benefits

The benefits of Automatic Text Summarization go beyond solving perceptible problems. Some other advantages of Text Summarization include:

Saves Time:

By generating automatic summaries, text summarization helps content editors save time and effort, which otherwise is invested in creating summaries of articles manually.

Instant Response:

It reduces the user's effort involved in exacting the relevant information. With automatic text summarization, the user can summarise an article in just a few seconds by using the software, thereby decreasing their reading time.

Increases Productivity Level:

Test Summarization enables the user to scan through the contents of a text for accurate, brief, and specific information. Therefore, the tool saves the user from the workload by reducing the size of the text and increasing the productivity level as the user can channel their energy to other critical things.

2. **AUTOMATIC MACHINE RECOGNITION OF FEATURES AND SENTIMENTS FROM ONLINE REVIEWS :**

E-commerce websites provide customers with the needed product information by giving a variety of services to choose from. One such service is to allow the customer to read the end user online reviews. Online reviews contain features which are helpful for the analysis in belief mining. Most of the systems work with the summarization of the features by taking the average features and their sentiments which leads to structured review information. Most of the times while classifying the sentiment of the review, the context of surrounding feature is undermined. In machine interpretable framework called Resource Description Framework (RDF) was introduce which helps in structuring these unstructured reviews in the form of features and sentiments obtained from traditional preprocessing and extraction techniques. The context data also supports for future ontology based analysis by taking the support of lexical database for word sense disambiguation. The Sentiments WordNet scores are used for sentiment word orientation. Many popular RDF vocabularies are helpful in the creation of such machine processable data. SQL queries are carried out on RDF data to learn the possibility for categorizing the reviews using feature information. This way to engineer the OWL Ontology for reasoning the RDF data. These results were processed by the interface as a feature, sentiment pair so that reviews are filtered clearly and help in satisfying the customer centric feature set.

3. **Data-driven Paradigm in Simplification** :

With the appearance of Simple English Wikipedia and its (comparable) alliance with English Wikipedia, which offered a large parallel dataset for training, It created opportunity for stronger NLP component of the systems and new challenges in text/sentence generation, but at the cost of blurring the final goal of those ATS systems, as there was no clear target population in mind anymore. The release of Newsela dataset (Xu et al., 2015) for English and Spanish in 2015, created opportunities for better modelling of simplification operations, given its well-controlled quality of manual simplifications at five different text complexity levels. Following the previously proposed idea of approaching ATS as a monolingual machine translation (MT) task (Specia, 2010; Coster and Kauchak, 2011), Xu et al. (2016) proposed an MT-based ATS system for English built upon Newsela and the large paraphrase database

(Pavlick and Callison-Burch, 2016). The manual sentence alignment of English Newsela (Xu et al., 2015), improved automatic alignment of EW-SEW corpus (Hwang et al., 2015), and the recently released free tools for sentence alignment (Paetzold et al., 2017; Stajner et al., ˇ 2017; Stajner et al., 2018), offered new opportunities for data-driven ATS. ˇ In 2017, several ATS systems explore various deep learning architectures appeared, using the new alignments of Wikipedia and Newsela for training. Sequence-to-sequence neural models (Nisioi et al., 2017; Stajner and Nisioi, 2018), and the neural model based on reinforcement learning techniques (Zhang ˇ and Lapata, 2017) showed a dominance of neural ATS approaches over the previous data-driven approaches in terms of quality of generated output (better grammaticality and meaning preservation). The question of simplicity of the generated output and the compliance of those models to different text genres and languages other than English, is still present. While solving the problems of grammaticality and meaning preservation, the neural TS systems introduced a new challenge, showing problems in dealing with abundance of name entities present both in news articles and Wikipedia articles.

Conclusion and Future work :

This method is good method for generating an automatic text generation. Since no model gives accurate result but our model provides better output and maximum output is accurate. Using our proposed model we have easily generated a fixed length and meaning full Bengali text.

There are some limitations this paper such as can not generate text without given the length of the text and n-gram sequence defined needed which is a lengthy process. Sometimes the order of the sentence is not correct in giving output. There are some defects in our proposed methodology such as can not generate random length text. We need to define the generating text length. Another defect is we need to define cushion token for predict next words. In our future work, we will make an automatic text generator which provides a random length hindi text without using any token or sequence.

References:

- Banko, Michele, Vibhu O. Mittal & Michael J.Witbrock. 2000. Headline generation based on statistical translation. Em Proceedings of the Annual Meeting of the Association for Compu-tational Linguistics, 318–325.
- Barzilay, Regina, Kathleen R McKeown & Mi-chael Elhadad. 1999. Information fusion in the context of multi-document summarization. EmProceedings of the Annual Meeting of the Association for Computer Linguistics, 550–557!
- Baxendale, Phyllis B. 1958. Machine-made index for technical literature: An experiment. IBM Journal of Research and Development 2(4).
- Berg-Kirkpatrick, Taylor, Dan Gillick & DanKlein. 2011. Jointly learning to extract and compress. Em Proceedings of the Annual Mee-ting of the Association for Computational Lin-guistics, 481–490.
- Carbonell, Jaime & Jade Goldstein. 1998. The use of MMR, diversity-based reranking for re-ordering documents and producing summaries.Em Proceedings of the Annual

International ACM SIGIR Conference on Research and Development in Information Retrieval, 335–336.

- Conroy, John M & Dianne P O'Leary. 2001.Text summarization via Hidden Markov Models. Em Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 406–407.
- Coster, William & David Kauchak. 2011. Learning to simplify sentences using Wikipedia.Em Proceedings of the Workshop on Monolin-gual Text-To-Text Generation, 1–9.
- Dorr, Bonnie, David Zajic & Richard Schwartz.2003. Hedge trimmer: a parse-and-trim approach to headline generation. Em Proceedings of the Conference of the North American Chapter of the Association for Computational Linguis-tics, 1–8.
- Edmundson, Harold P. 1969. New methods in au-tomatic extracting. Journal of the ACM 16(2).Erkan, G¨unes & Dragomir R Radev. 2004. Lex-Rank: Graph-based lexical centrality as sali-ence in text summarization. Journal of Artifi-cial Intelligence Research 2(1).Franceschet, Massimo. 2011. PageRank: standing on the shoulders of giants. Communications of the ACM 54(6).
- Fung, Pascale & Grace Ngai. 2006. One story, one flow: Hidden Markov story models for multilingual multidocument summarization. ACM Transactions on Speech and Language roces-sing 3(2).