Comparative Study of Text Summarization Methods:

Text summarization is a process to express the content of a document in a condensed form that meets the needs of the user. Summarization is a two-step process. The first step is the extraction of important concepts from the source text by building an intermediate representation of some sort and the second step uses this intermediate representation to generate a summary. The paper discusses in detail two main categories of text summarization methods these are extractive and abstractive summarization methods. One of the approaches for summarization can be done by sentence extraction and clustering. The method used to cluster the sentences is k-means algorithm [2].

Due to rapid growth of technology and use of Internet, there is information overload. This problem can be solved if there are strong text summarizers which produces a summary of document to help user. One possible solution is to summarize a document using either extractive or abstractive methods.

Summarization by extractive just extracts the sentences from the original document and adds them to summary. Extractive method is usually easy to implement and is based on statistical features not on semantic relation with sentences and it tends to be inconsistent. Summarization by abstractive generates a sentence from a semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. It provides more generalized summary but it is difficult to compute. Abstractive and extractive summarization uses either statistical or linguistics approaches or combination of both to generate summary.

Various summarization methods can be compared based on the type of summary and application. Summarization system can be classified based on approaches, based on type of details, based on type of content, based on limitation, based on number of input documents and based on language.

Statistical approaches [3] can summarize a document using statistical features of the sentence like title, location, term frequency, assigning weights to the keywords and then calculating the score of the sentence and selecting the highest scored sentence into the summary. Importance of a sentence can be decided by several methods such as title method, location method, tf-idf method, cue word method.

Linguistic is a scientific study of language which includes study of semantics. Abstractive text summarization is based on linguistic method which involves the semantic processing for summarization. Miller et al proposed strong concepts with the help of linguistic features but they require much memory for saving the linguistic information like Word Net and processor capacity because of additional linguistic knowledge and complex linguistic processing. Various methods are Lexical chain, word net, graph theory and clustering.

This paper discussed different types of summarization methods used for summarizing a document and advantages and disadvantages of each method

iTiger: An Automatic Issue Title Generation Tool

In software development and maintenance, bug reports or issues are heavily used by developers to report bugs or propose new features. Prior research [5] finds that well-written bug reports are more likely to gain triager's attention and influence the decision on whether the bugs get fixed. There is an emerging research interest in improving issue quality. An issue usually includes a title and a description. The title serves as the summary of the description. However, since developers may neglect to compose a succinct and accurate issue title, there is a need for an automatic issue title generation tool to help developers.

In this work the first attempt was to fine-tune BART, which has been pre-trained using English corpora, to generate issue titles. The fine-tuned BART is implemented as a web tool named iTiger, which can suggest an issue title based on the issue description.

Chen et al. [6] are the first to work on the issue title generation task, which aims to help developers write issue titles. They formulated the issue title generation task as a one-sentence summarization task. They proposed iTAPE, which is a specialized tool to generate the issue title based on the issue description. iTAPE relies on a sequence-to-sequence model. Fine-tuning PTMs can usually perform better than learning models from scratch. To fill the gap of adopting PTMs to solve the issue title generation task a type of PTM.

BART is fine-tuned by feeding it with the pairs of issue descriptions and titles. In the inference stage, the input is the issue description and BART can generate the issue title as the output. BART [7] is used which has demonstrated promising performance in summarization and text generation tasks. BART model used has been pre-trained in large English corpora. Once the model is fine-tuned, it can be utilized to generate the suggested title based on the issue description.

iTigger does not focus on bug reports and focuses only on a part of issues. A difference between Bug report summarization and issue title is the length of the target sequence: bug report summaries usually contain several sentences, whereas an issue title is a single-sentence summary.

In this paper, iTiger is presented which generates the issue title based on the issue description. iTiger allows developers to modify the generated titles. iTiger utilizes the state-of-the-art summarization model, i.e., BART. iTiger is easy to use and useful. The evaluators are also willing to use this tool in their daily work.

Automatic summarising: The state of the art

In this overview of paper [8], the authors examine the evaluation strategies applied to summarizing, the issues they raise, and the major programmes. They discuss input, purpose, and output factors, as well as extractive and non-extractive strategies.

Automatic summarisation has made significant progress, with useful applications, better evaluations and better task understanding. But summarizing systems still lack motivation in relation to factors affecting them.

NLP can deliver useful summaries like parsing. Sparck Jones assumes a tripartite processing model distinguishing three stages. The interpretation of a source text into a source representation, its transformation into a summary representation, and its generation into a summary text. The definition and model are deliberately broad enough to support processing of summaries of many different kinds.

NLIP system evaluation, including summary evaluation, has been applying the distinction between intrinsic and extrinsic evaluation. The former refers to the extent to which a system meets its own objectives, the latter to its functional effectiveness in context. Experience with summary evaluation suggests the intrinsic/extrinsic distinction is too crude and that a finer granularity is needed. Evaluation without any reference to purpose is of extremely limited value.

The design and evaluation of summarising systems have to be related to the three classes of context factor. Input factors characterising the source material including style and units, purpose factors including intended use and audience, and output factors including reduction and format.

Summaries do not have to consist of running text: they may be phrase lists or tables with slot phrase fillers. But as running text is commonly required. Unfortunately, text quality is too weak to be a system discriminator. Direct evaluation of content capture implies source markup for important content and summary inspection

In relation to summarising techniques themselves, this wave of work has been useful in exploring the possibilities and potential utilities of extractive summarising. A domain ontology is an example of a statistical or shallow symbolic method that doesn't require heavy model instantiation. These techniques can deliver useful results when the summary requirements are modest, and hybrid techniques a bit more than purely statistical techniques.

REFERENCES:

[1]https://www.researchgate.net/profile/Nikita-Munot/publication/284411761_Comparative_Study_of_Text_Summarization_Methods/links/5cc049d492851c8d2200ad99/Comparative-Study-of-Text-Summarization-Methods.pdf

[2] ZHANG Pei-ying, LI Cun-he. Automatic text summarization based on sentences clustering and extraction.

[3] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, Bahareh Gholamzadeh.A Comprehensive Survey on Text Summarization Systems. 2009 In proceeding of: Computer Science and its Applications, 2nd International Conference.

[4] https://arxiv.org/pdf/2206.10811.pdf

[5] Philip J Guo, Thomas Zimmermann, Nachiappan Nagappan, and Brendan Murphy. 2010. Characterizing and predicting which bugs get fixed: an empirical study of microsoft windows. In Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering-Volume 1. 495–504.

[6] Songqiang Chen, Xiaoyuan Xie, Bangguo Yin, Yuanxiang Ji, Lin Chen, and Baowen Xu. 2020. Stay professional and efficient: Automatically generate titles for your bug reports. In 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 385–397.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, 7871–7880.

[8]https://www.sciencedirect.com/science/article/pii/S0306457307000878?casa_token=uFIBfDObhmEAAAAA:Mvn51Goa70w9Q6hMCqNYH4tXl-IYd0Ngz8trdvLRJJftgDPg2oG880EhtGGAPirO5Logt7h4Yjc_