Automatic Summarization of Legal Bills: A Comparative Analysis of Classical Extractive Approaches

Deepali Jain
CSE Department,
National Institute of Technology Silchar,
Assam-788010, India
jaindeepali010@gmail.com

Malaya Dutta Borah
CSE Department,
National Institute of Technology Silchar,
Assam-788010, India
malayaduttaborah@cse.nits.ac.in

Anupam Biswas
CSE Department,
National Institute of Technology Silchar,
Assam-788010, India
anupam@cse.nits.ac.in

2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) | 978-1-7281-8529-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCIS51004.2021.9397119

***Abstract*—A Legal Document is typically very long and structurally rich, which makes a quick understanding of the document very difficult. One way to deal with this difficulty is to manually summarize these documents with the help of legal experts. However, this is a costly and time consuming process. Design of automatic summarization approaches could be the key, to achieve a more time and cost effective solution to this problem. In this work, a detailed comparative analysis of multiple classical extractive summarization techniques is presented, with respect to Recall-Oriented Understudy for Gisting Evaluation metrics (ROUGE), Bilingual evaluation understudy metrics (BLEU) and Cosine Similarity scores, which are widely used metrics for evaluating automatic summarization approaches. The experimental analysis is performed on a publicly available benchmark dataset. From the experimental results, it has been observed that graph based summarization techniques perform well in general, across all the evaluation metrics. Another important observation is that, in addition to word frequency, considering other key contextual information can boost the performance of automatic summarization techniques. This comparative analysis work can serve as a baseline on the benchmark legal dataset, which is expected to be helpful for further research in this domain.**

***Keywords*—legal document summarization; legal extractive summarization; comparative analysis; legal domain**

I. Introduction

In this age of data deluge, a large amount of online data is being produced each minute of every day. This sort of large amount of information is also available online, in the field of law in the form of legal documents. A document is legal if the intention behind creating it is enforcement in the court of law. These documents are quite long in terms of its structure as compared to a general document, because of which it becomes difficult to read and understand. It would be better if shorter versions are available for these long documents in the form of summary. A good summary is a shorter version of any document, which contains all the relevant information from the document. This can be achieved by the technique known as automatic text summarization, whose task is to produce summaries without losing relevant information. It has been seen that such techniques have high utility in the field of law as well, which has led to the introduction of Automatic Legal Document Summarization Domain– a sub-domain of text summarization in general. Currently, a large number of legal experts are involved in generating legal summaries, which is time consuming and requires a considerable amount of human efforts. For example, legal professionals like judges and lawyers need to send their cases to legal experts for producing summaries. The novice readers also want to get an idea of the current cases as well as previous similar related cases without going through long documents. Now-a-days, judgement cases are often easily available through online sources like The Judgment Information System [1], Indian Kanoon [2], AustLII [3], etc. so that an ordinary citizen can also access them. Therefore, automatic summarization tools are very helpful for both legal practitioners as well as for ordinary citizens, since with the help of such systems, summaries of any cases will become easily accessible. Such types of tools will also lead to transparency, because a lot of hard legal jargon which is difficult to understand, can be easily avoided.

It is important to note here that, most of the work done in the area of legal document summarization, has solely focused on experimental analysis on privately collected datasets, which makes it hard to provide a fair and detailed comparison of various summarization techniques.

In this work, a comprehensive comparative analysis of multiple classical extractive techniques like Luhn, Edmundson, Latent Semantic Analysis (LSA), Reduction, Textrank, Lexrank, Kullback-Leibler (KL), Sumbasic is presented, with extensive experimental evaluation on the BillSum dataset [4] which is a publicly available benchmark dataset for legal document summarization. From the comparative analysis, it has been found that graph based approaches perform better than word frequency based approaches in general; however, when other key contextual information is used along with word frequency, good summarization performance can be achieved.

The rest of the paper is organized as follows: in Section 2, an investigation about several previous text summarization work in general and legal specific text summarization work is presented. In Section 3, a detailed description of the proposed comparative analysis methodology is presented. Following which, Section 4 presents the experimental results on the BillSum Dataset, which are then discussed in Section 5. Finally, in Section 6 the paper is concluded with the key findings of this work and a brief discussion on future research directions.

II. Related Work

In the domain of text mining, there are two types of summaries that a machine can generate: 1) Abstractive and 2) Extractive. Abstractive summaries are more like human written summaries– summaries which humans write in their own words while extractive summaries comprise important sentences picked from the corresponding document. Extractive summaries can be thought of as highlighting the important contents on a written document. Initially, for text summarization in general, several classical extractive approaches had been proposed, that are based on frequency analysis of words [5] – [9]. Various graph based approaches are also proposed for extractive summarization, which in general perform well on text summarization tasks [10] – [13]. One of the good works based on vector space models is also proposed for extractive summarization [14]. With the recent advancements in the field of deep learning (DL), various extractive summarization techniques have also been recently proposed which make use of DL techniques [15], [16]. Several works have also been done for abstractive summarization using recurrent neural networks (RNN) based DL approaches, reinforcement learning and pre-training models [17] – [20].

However, due to the large size and rich structure of legal documents and their summaries, such abstractive approaches have been shown to take a considerably large amount of training time and do not typically perform well in the summarization task [21]. In the legal domain, several works have been done in the past which aim to create effective legal document summaries [22] – [25]. Many legal systems have also been proposed over the past years which summarizes legal documents such as Letsum [26], Kaftie [27], HAUSS [28], CaseSummarizer [29].

III. Methodology

The methodology primarily consists of four major phases as shown in Fig. 1.

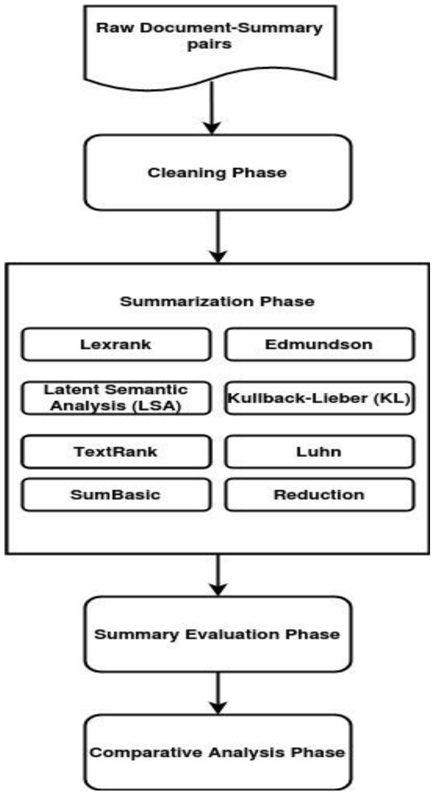

Fig. 1. Methodology of the work

A. Cleaning Phase

In this cleaning phase, multiple data preprocessing steps have been taken in order to prepare the dataset for the further phases. More specifically normalization over certain words and sentences, get rid of semicolons, remove any white spaces, attempt to make one complete sentence by removing bullet to finally form a paragraph, remove the annoying special characters, also make sure that there is a space between periods and start of the new sentence is carried out so that the documents and summaries are more suitable for performing summarization. Similar cleaning steps have been performed by Eidelman [4] who first introduced the BillSum dataset, which is a publicly available legal benchmark dataset consisting of United States (US) Congressional and California (CA) state bills.

B. Summarization Phase

In this phase, several classical extractive summarization algorithms are applied. Various graph based algorithms like Textrank, Lexrank, reduction algorithms based on the concept of Google's Pagerank algorithm are used, some heuristic based algorithms like Luhn, Edmundson and some statistical algorithms like KL[9], Latent Semantic Analysis algorithms and SumBasic. In the Textrank algorithm [11], the text units which best describes the task is represented as vertices. Based on some 'similarity relation', a link is drawn from one vertex to another. These links are called edges. Then, a ranking is done based upon the final scores attached with each vertex and sort them. Another graph based algorithm Lexrank[13] which is an unsupervised approach in which the importance of sentences is based upon the eigenvector centrality based upon which scoring of sentences is done. Yet another graph based algorithm is reduction algorithm[10] which is based on the idea that some extraneous phrases need to be removed from the sentence and thus form a summary. The whole reduction process is based on the 'Graph Reduction' idea. In order to reduce the graph, some context information is considered. Importance of word is determined by the lexical links between words and scores of such words are then calculated. The score of the sentence is determined by calculating the score of the individual words. The score tells the importance of a sentence in its local context.

HP Luhn introduced a heuristic approach of summarizing the documents based on word frequency and position of that word [5]. Edmundson had used more number of features in addition to the feature used by Luhn. The author has considered four features: Cue, Title, Keyword and Location for determining the importance of a sentence. All these features are taken in the form of linear combination. Latent Semantic Analysis [7] is the technique which identifies the semantically important sentences. The approach is based on Singular Value Decomposition (SVD). Kullback-Leibler (KL) Divergence [9] is a sentence selection algorithm which greedily adds sentences into the summary until KL Divergence decreases. SumBasic [8] is a summarization algorithm whose working is based on the idea that most frequent words in a document are more important for summary formation than the less frequent words.

C. Summary Evaluation Phase

In this phase, scores of the predicted summaries with respect to the actual summaries are calculated. For this, three measures are considered as mentioned below.

- ***ROUGE*** or Recall-Oriented Understudy for Gisting Evaluation metrics [30] is a recall based measure which measures the n-gram overlap between the generated summary and the reference summary. There are several variants of rouge metrics such as ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-N, ROUGE-L. In this work, ROUGE-1, ROUGE-2 and ROUGE-L are considered. ROUGE-1 is the overlapping of unigrams whereas overlapping of bigrams is known as ROUGE-2. ROUGE-L identifies the longest common subsequence co-occurring in n-grams. The definition for ROUGE-N is given by Eq. (1).

$$R_N = \frac{\sum_{S \epsilon RS_{set}} \sum_{gram_n \epsilon S} Count_{match}(gram_n)}{\sum_{S \epsilon RS_{set}} \sum_{gram_n \epsilon S} Count(gram_n)} \quad (1)$$

where R_N is ROUGE-N, S is a sentence, RS_{set} are the reference summaries $gram_n$ and $count_{match}$ is the maximum number of n-grams that matches in a candidate summary and a set of reference summaries, n is the length of n-gram. Here N stands for n- gram's length.

- ***BLEU***or Bilingual Evaluation Understudy metrics [31], which is a precision based measure compares a candidate summary against a reference summary. The n-gram precision is given by Eq. (2).

$$P = \left\{ \prod_{i=1}^{N} P(i) \right\}^{1/N} \quad (2)$$

where N is N-gram order whose value is 4.

Brevity Penalty, *BP*, is introduced to penalize short hypotheses which is given by Eq. (3).

$$BP = exp\left\{ min\left(0, \frac{h-R}{h}\right)\right\} \quad (3)$$

where h is the hypothesis length and R is the reference length.

In this way, BLEU score is n-gram precision times brevity penalty which is given by Eq. (4).

$$BLEU = BP \times P \quad (4)$$

- ***Cosine Similarity***measures how similar the two summaries are. In this work, an actual and predicted summary is represented into a 50-sized vector using word2vec model in which element wise averaging across words and sentences is done. Then similarity is calculated between the actual summary and predicted summary of each BillSum document.

D. Comparative Analysis Phase

In this phase, various classical extractive text summarization techniques are compared with respect to ROUGE-1, ROUGE-2, ROUGE-L, BLEU and Cosine Similarity scores on the US Training, US Test and CA Test subsets of the BillSum dataset.

IV. Experimental Results And Analysis

All the summarization algorithms which are discussed in Section 2 are applied to the 18,949 US BillSum training, 3,269 US BillSum test and 1,237 CA BillSum test dataset. The experiments were performed using Google Collaboratory which is a free online cloud-based Jupyter notebook, with the help of freely available Python based NLP packages: Gensim [32] and Sumy [33]. The experimental results using the algorithms on US BillSum training data, US BillSum testing data and CA BillSum testing data are shown in Fig. 2 through Fig. 10. The size of the summaries to be generated by the algorithms is determined using the average summary to document ratio of the US BillSum training dataset, which is found to be 0.2.

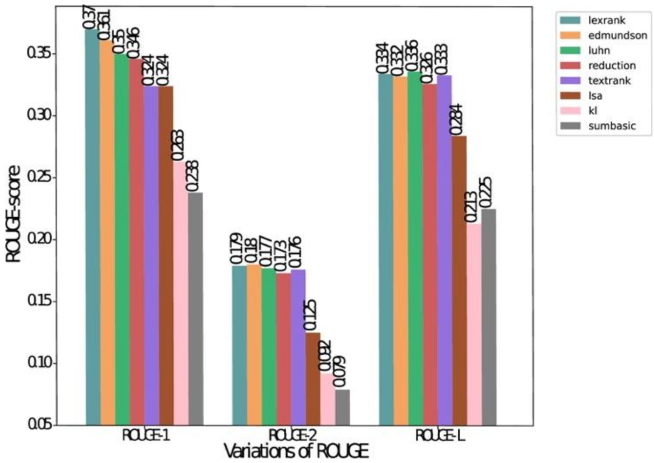

Fig. 2. ROUGE scores on US training data.

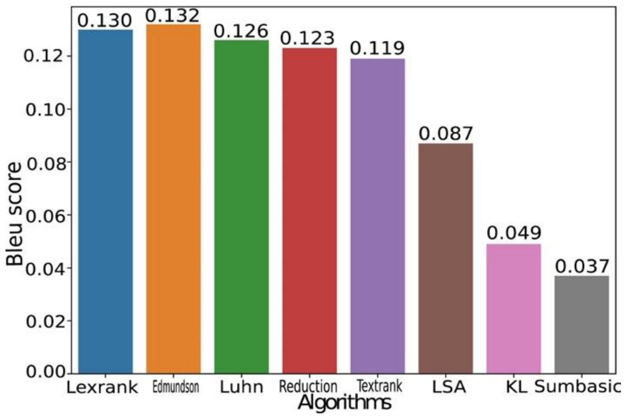

Fig. 3. BLEU scores on US training data.

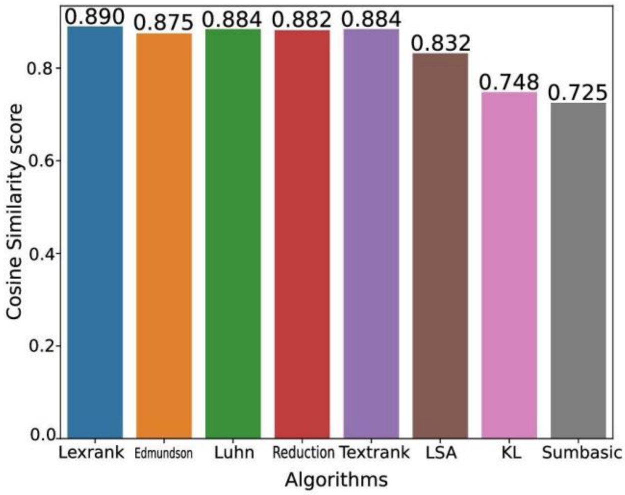

Fig. 4. Cosine Similarity scores on US training data.

Fig. 2, 3 and 4 shows ROUGE score, BLEU score and cosine similarity score respectively on US BillSum Training dataset. Lexrank achieves the best ROUGE-1 score and cosine similarity scores of 0.370 and 0.890 respectively whereas both Lexrank and Edmundson achieves the best Rouge-2 score of 0.180. Luhn has the highest ROUGE-L score of 0.336. In terms of BLEU score, Edmundson achieves the highest score of 0.133 among all the algorithms.

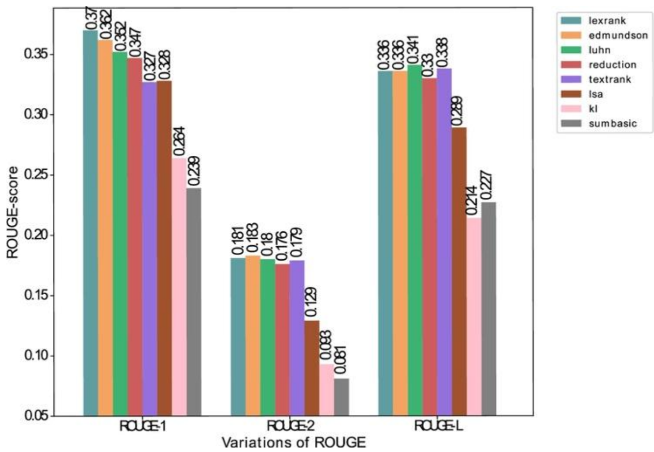

Fig. 5. ROUGE scores on US testing data.

Fig. 5, 6 and 7 shows the results on US BillSum Testing Dataset. Lexrank achieves 0.370 ROUGE-1 score and 0.890 cosine similarity score which is the highest among all other algorithms. Edmundson achieves highest ROUGE-2 score and BLEU score of 0.183 and 0.133 respectively whereas Luhn has highest ROUGE-L score of 0.341.

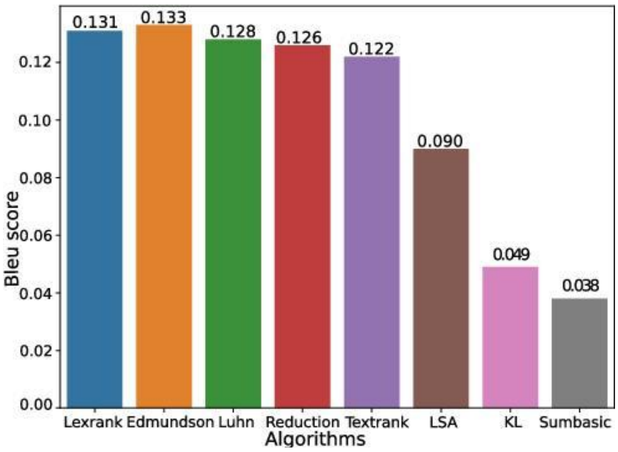

Fig. 6. BLEU scores on US testing data.

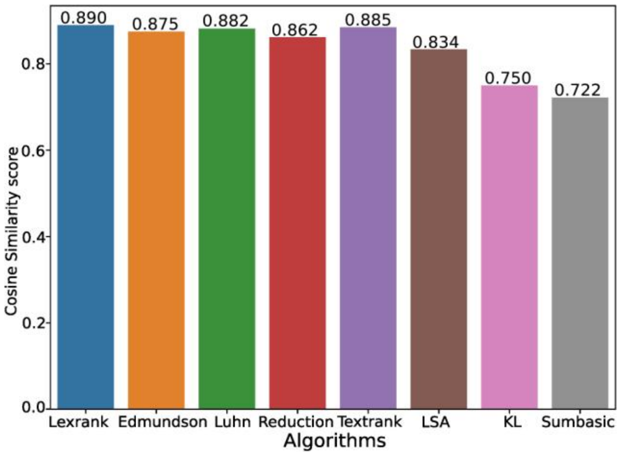

Fig. 7. Cosine Similarity scores on US testing data.

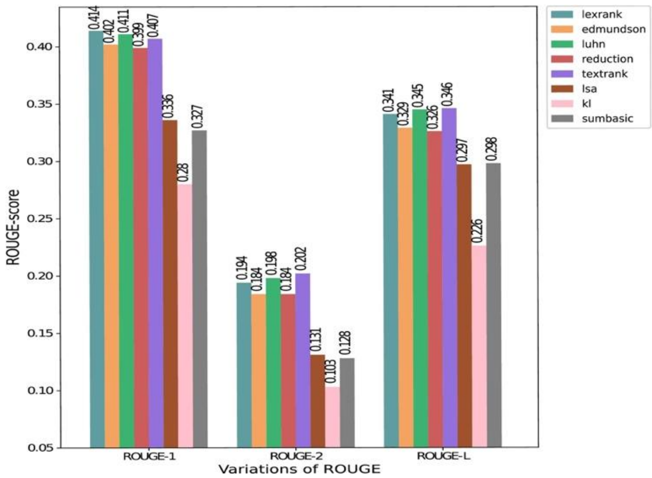

Fig. 8. ROUGE scores on CA testing data.

Fig. 8, 9 and 10 shows the results on CA BillSum Testing Dataset. Lexrank achieves 0.414 ROUGE-1 score which is the best among all other algorithms whereas Luhn achieves best BLEU score of 0.156 and Textrank achieves best ROUGE-2, ROUGE-L and Cosine Similarity score of 0.202, 0.346 and 0.931 respectively.

According to the current state of the art, in the domain of text summarization, the results that are typically obtained are in the range of 0.38 to 0.42 for ROUGE-1, 0.15 to 0.20 for ROUGE-2, and 0.32 to 0.38 for ROUGE-L [34]. The results that have obtained in this work also follows the same trend.

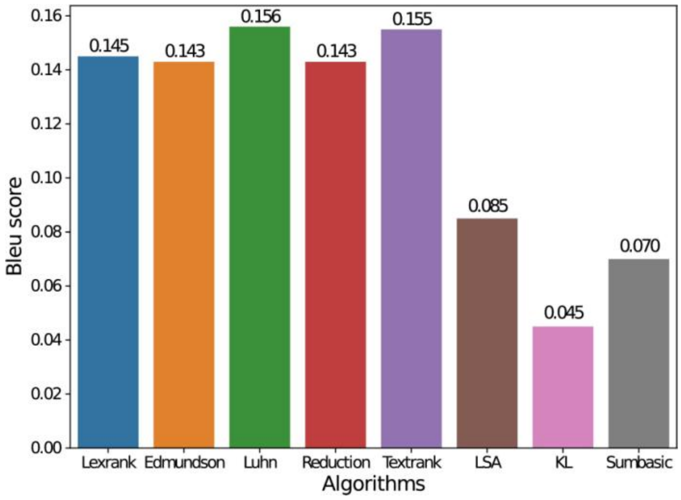

Fig. 9. BLEU scores on CA testing data.

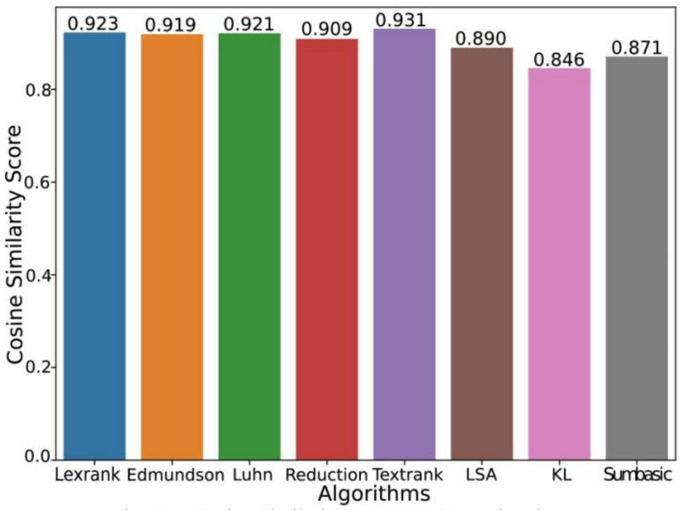

Fig. 10. Cosine Similarity scores on CA testing data.

From these results, one of the trends that has been observed is that, the algorithms which performed good are good for all the three cases, and those which performed bad remains bad for all the three cases. More specifically, Lexrank, Edmundson, Luhn, Textrank and Reduction have consistently performed better, for all performance metrics; while LSA, KL and SumBasic have performed poorly in almost all the cases. The reason for the poor performance of LSA, KL and SumBasic approaches is that, all these algorithms are word frequency based algorithms, it means that these algorithms make use of limited information i.e. only word information for summarization purpose. So, it is natural for these algorithms not to perform well as compare to other algorithms which are more sophisticated. The results have shown that Lexrank, which is a graph based algorithm has performed better in terms of ROUGE-1 and Cosine Similarity score. The ROUGE-2 score of Lexrank is also comparable with those algorithms whose ROUGE-2 is the best. The reason behind its good performance is that, the algorithm considers the eigenvector centrality, of a sentence graph that only consists of edges with high cosine similarity, for finding important sentences. Moreover, it also considers the prestige of voting nodes in weighting each vote while finding eigenvector centrality of the graph. It has also been observed that the performance of Edmundson is good in terms of ROUGE-2 and BLEU score. This is primarily due to the fact that the algorithm considers many other features in addition to keyword frequency, such as, Cue words, Title, Location, etc., thereby increasing the amount of information used for the purpose of summarization. The performance of Luhn is experimentally found to be better in terms of ROUGE-L score, because it considers the relative position of significant words in addition to the frequency of these significant words. This has a significant impact on ROUGE-L score, because ROUGE-L is the longest common subsequence based metric for which position and context related information is very important.

The most widely used type of evaluation metric for text summarization is the ROUGE-N metric, however, there are other popular approaches for comparing a candidate text with a reference text. In order to explore such approaches for the summarization tasks, in this work, BLEU scores and cosine similarity measures are also used. BLEU is a precision- based measure, whose value decreases with the increase in length of sentences. In our work, the legal documents as well as their reference summaries are long, because of which BLEU scores are low in case of every algorithm. Cosine similarity is a vector-similarity based approach which can be used for text comparison when combine with some form of text vectorization approach. From the experimental results, it is clear that, the cosine similarity based scores also follow the same trend as ROUGE and BLEU scores.

V. Discussion

From experimental results, it has been shown that SumBasic, KL and LSA algorithms performed poorly, across all the datasets for all evaluation metrics. All of these algorithms are based on frequency analysis of words which only makes use of limited information for summarization. For example, SumBasic considers the probability distribution of each word in a sentence, assigns a weight to sentences and picks the best scoring sentences whose word probability is the highest, while KL considers the unigram distribution of words and adds sentences to summary until KL divergence decreased. The first step of LSA involves forming a Document- Term matrix which is also based on the word information. All of these techniques consider distribution of words in a unigram fashion and do not consider any other aspect like context relationship with other words. Hence this is the reason why all of these techniques have low scores as compared to techniques like Luhn and Edmundson which also considers relative importance of word in its context and some other features in addition to word frequency respectively. Since, Edmundson took into account some other features like Cue, Title, Location in addition to word frequency which Luhn considered, it is obvious for the Edmundson algorithm to perform better than Luhn as it is seen from the experimental results given in Fig. 2 through Fig. 10.

It can be observed from the experimental results that Graph based approaches like Textrank and Lexrank performed even better than frequency based approaches. One key characteristic of graph based approaches is that, they consider sentence similarity and not just depend only upon words. They also make use of concepts like Eigenvector centrality [13] which made these algorithms extremely powerful for ranking sentences. One of the algorithm which is based on Graph Reduction idea [10] also performed better than frequency based approaches. The main idea of this algorithm is to remove unnecessary phrases from the sentence under consideration. It removes sub-trees from a tree based on the different type of information being collected at the leaf nodes. The algorithm considers syntactic knowledge, context and probability information for deletion of extraneous phrases that made it perform better than simple frequency based ideas.

There are two main findings of this comparative analysis work where the performance of classical extractive summarization techniques has been explored on publicly available benchmark legal summarization dataset.

- In general, graph based approaches perform better than simple word-frequency based approaches, because they consider sentence level relationships also.
- Only word frequency information is not sufficient for effective summarization, since it ignores many other important features like syntactic knowledge, context information, lexical information, etc.

One important thing to note here is that, this work only considers the classical extractive summarization approaches applied in the legal domain. However, two of the more recent approaches, supervised extractive summarization and deep learning based abstractive summarization, need further exploration in this domain. The applicability of machine learning and deep learning based supervised extractive techniques need to be explored in the legal domain, which can make use of the available supervisory signals from the actual summaries that are typically available in benchmark datasets. Furthermore, the area of deep learning based abstractive summarization is a relatively under-explored area. This is due to the fact that abstractive summarization approaches do not perform as well as extractive summarization techniques when the size of the texts involved is large, which is typically the case with legal documents. However, the recent advancements in the field of abstractive summarization needs to explored further in the legal domain as well.

VI. Conclusion And Future Work

A comprehensive comparative analysis of multiple classical extractive text summarization techniques have been presented in this work, with experimental evaluations on the publicly available BillSum benchmark dataset. From experimental observations it can be concluded that graph-based summarization techniques perform well on legal document summarization tasks in general. Also, considering only word frequency related information is not found to be the correct strategy for an effective summarization; instead other key information like syntactic knowledge, context information, lexical information, etc. may help improve the automatic summarization systems' performance significantly.

The experimental results from this work can be considered as baselines for carrying out further explorations in this area. One of the key sets of techniques that needs further study is the set of supervised learning based techniques, that can make use of the reference summaries in order to learn better predictive models for extractive summarization. The design of efficient DL based abstractive summarization techniques for legal document summarization is another under explored area. The study of these machine learning based summarization ideas on publicly available benchmark datasets will be part of the future work.

References

[1] The Judgement Information System: http://164.100.79.153/judis/ Last accessed on: November 19, 2020.

[2] Indian Kanoon: https://indiankanoon.org/ Last accessed on: November 19, 2020.

[3] Australasian Legal Information Institute: http://www.austlii.edu.au/ Last accessed on: November 19, 2020.

[4] Eidelman, V. (2019, November). Billsum: A corpus for automatic summarization of us legislation. In Proceedings of the 2nd Workshop on New Frontiers in Summarization (pp. 48-56).

[5] Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM Journal of research and development, 2(2), 159-165.

[6] Edmundson, H. P. (1969). New methods in automatic extracting. Journal of the ACM (JACM), 16(2), 264-285.

[7] Steinberger, J., &Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. Proc. ISIM, 4, 93-100.

[8] Nenkova, A., &Vanderwende, L. (2005). The impact of frequency on summarization. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005, 101.

[9] Haghighi, A., &Vanderwende, L. (2009, June). Exploring content models for multi-document summarization. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 362-370).

[10] Jing, H. (2000, April). Sentence reduction for automatic text summarization. In Sixth Applied Natural Language Processing Conference (pp. 310-315).

[11] Mihalcea, R., &Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[12] Erkan, G., &Radev, D. (2004, July). Lexpagerank: Prestige in multi-document text summarization. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (pp. 365-371).

[13] Erkan, G., &Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, 22, 457-479.

[14] Kågebäck, M., Mogren, O., Tahmasebi, N., &Dubhashi, D. (2014, April). Extractive summarization using continuous vector space models. In Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) (pp. 31-39).

[15] \Nallapati, R., Zhai, F., & Zhou, B. (2017, February). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence.

[16] Narayan, S., Cohen, S. B., &Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. arXiv preprint arXiv:1802.08636.

[17] Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685.

[18] Paulus, R., Xiong, C., &Socher, R. (2017). A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304.

[19] Gehrmann, S., Deng, Y., & Rush, A. M. (2018). Bottom-up abstractive summarization. arXiv preprint arXiv:1808.10792.

[20] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2019). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. arXiv preprint arXiv:1912.08777.

[21] Bhattacharya, P., Hiware, K., Rajgaria, S., Pochhi, N., Ghosh, K., & Ghosh, S. (2019, April). A comparative study of summarization algorithms applied to legal case judgments. In European Conference on Information Retrieval (pp. 413-428). Springer, Cham.

[22] Saravanan, M., Ravindran, B., & Raman, S. (2006). Improving legal document summarization using graphical models. Frontiers in Artificial Intelligence and Applications, 152, 51.

[23] Saravanan, M., Ravindran, B., & Raman, S. (2008). Automatic identification of rhetorical roles using conditional random fields for legal document summarization. In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.

[24] Galgani, F., Compton, P., & Hoffmann, A. (2012, September). Citation based summarisation of legal texts. In Pacific Rim International Conference on Artificial Intelligence (pp. 40-52). Springer, Berlin, Heidelberg.

[25] Galgani, F., Compton, P., & Hoffmann, A. (2012, April). Combining different summarization techniques for legal text. In Proceedings of the workshop on innovative hybrid approaches to the processing of textual data (pp. 115-123).

[26] Farzindar, A., & Lapalme, G. (2004). Letsum, an automatic legal text summarizing system. Legal knowledge and information systems, JURIX, 11-18.

[27] Pham, S. B., & Hoffmann, A. (2004, December). Incremental knowledge acquisition for building sophisticated information extraction systems with kaftie. In International Conference on Practical Aspects of Knowledge Management (pp. 292-306). Springer, Berlin, Heidelberg.

[28] Galgani, F., Compton, P., & Hoffmann, A. (2014). HAUSS: Incrementally building a summarizer combining multiple techniques. International journal of human-computer studies, 72(7), 584-605.

[29] Polsley, S., Jhunjhunwala, P., & Huang, R. (2016, December). Casesummarizer: a system for automated summarization of legal texts. In Proceedings of COLING 2016, the 26th international conference on Computational Linguistics: System Demonstrations (pp. 258-262).

[30] Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries ACL. In Proceedings of Workshop on Text Summarization Branches Out Post Conference Workshop of ACL (pp. 2017-05).

[31] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

[32] R&Sojka, P. (2010). Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

[33] Sumy: Automatic Text Summarizer https://github.com/miso-belica/sumy Last accessed on: November 20, 2020.

[34] NLP-Progress: Text Summarization State-of-the-art. http://nlpprogress.com/english/summarization.html Last accessed on: November 20, 2020.