**LexRank and PEGASUS Transformer for Summarization of Legal Documents**

Sarthak Dalal1[0000-0003-0517-9831], Amit Singhal2[0000-0002-4010-6614] and Brejesh Lall3[0000-0003-2677-3071]

1 Dwarkadas J Sanghvi College of Engineering, Mumbai 400056, India

2 Netaji Subhas University of Technology, Delhi 110078, India

3 Indian Institute of Technology, Delhi 110016, India

**Abstract.** Legal documents are generally verbose and contain lots of dense legal text. Lawyers often must study prior cases and reading such documents can be time-consuming. Such documents may also be incomprehensible to the ordinary public who lack legal understanding. Enormous amounts of legal data online have made access to case files and documents simple. Hence, automatic summarization and paraphrasing using Machine Learning (ML) of such documents has become an important area of research to make the documents comprehensible for lawyers and the ordinary public. In this paper, we review previous approaches to automatic summarization and compare their output on six different documents. We also provide a summarization and paraphrasing technique using the LexRank and PEGASUS transformer for abstractive summarization. The summary produced by the LexRank model outperforms the other models tested by obtaining a higher ROUGE score. The final summary retains the important information from the source document and is paraphrased to a simpler language.

**Keywords:** Natural Language Processing, Machine Learning, LexRank, PEGASUS

1. Introduction

India is a country that follows the *Common Law* system [1]. The fundamental feature of common law is that it uses *Precedents* [2] in circumstances where the parties disagree on what the law is. In such situations, the court looks to previous precedential decisions of competent courts and reconstructs the principles of those previous cases as applicable to the current facts. Thus, precedents aid a lawyer in preparation for a case by allowing them to learn how the court has handled similar matters in the past. As a result, they must analyze a large number of previous cases. To study previous cases, the lawyer must go through law reports/case judgements, which are generally verbose and contain a lot of dense legal text. Even for a legal professional, reading and comprehending the complete text of a case is challenging. Summaries of case judgements are particularly useful in situations like these. Text summarizing condenses the material of a document without sacrificing its meaning in order to save consumers' time and cognitive strain. From the perspective of a common citizen, these documents are crammed with legal jargon, and even comprehending a summarized text might be arduous. Hence, it is necessary to paraphrase documents along with summarization for the common public.

Presently, the process of summarization of case documents is slow, laborious and expensive as documents are summarized manually by legal experts. Using ML can accelerate this process and summarize large amounts of documents rapidly in a cost-effective manner. Ordinary citizens generally do not have any access to legal experts to summarize the documents for them and hence, using ML is the only way they can comprehend a legal document. If we have approaches that automate the process of writing case summaries as well as paraphrasing them into simpler language using ML, the user's freedom to examine legal material is considerably increased. The user may choose cases based on her preferences without the intervention of attorneys, who might hide many cases from the user. In this paper, we present a tool for unsupervised summarization and paraphrasing for legal documents. The documents are summarized using the *LexRank* method which is an unsupervised graph-based approach for extractive text summarization. Then, the summary is paraphrased using the *PEGASUS* transformer. The final output is an abstractive summary of the original document

1. Related work

Lots of work has been done in general summarization techniques but summarizing legal documents is a relatively unexplored avenue. In order to study previous works under this domain, we summarized six documents using the methods under this section. Apart from considering ROUGE scores, we manually surveyed other factors of the summary, such as the length, comprehensibility, grammar and punctuations, etc.

* 1. Summarization for legal documents

There have been works for developing models specifically to summarize legal documents using ML which have employed both supervised and unsupervised Machine Learning approaches. In [3], the authors used probabilistic graphical models to summarize documents of legal domain. The rhetorical roles of a sentence were identified using *Conditional Random Fields.* The automatic summarization process starts by preprocessing the documents into segments, sentences, and tokens.

Feature identification techniques in this paper included the understanding of abbreviated texts, section numbers and arguments specific to the structure of legal documents. K-mixture model is used as the term distribution model to assign probabilistic weights. Sentences are then re-ranked twice, first based on their weights and again based on their evolving roles throughout CRF implementation, to provide the final summary. While reproducing this summarization model, all the six summaries were around the same size, irrespective of their source document. Because of the sentence rankings, sometimes the summary does not seem to make proper sense.

A supervised model has its downsides as the model can only give accurate results for the case documents of the court it was trained on. For instance, the Graphical Model described above was trained on the documents of Kerala High Court while we tested our implementations on documents of the Indian Supreme Court (which is an Indian court but still gives less accurate results). If the disparity is higher, for instance, if a model is trained on the Australian court and made to summarize documents of the Indian Supreme court, the results would be unintelligible. In such cases, an unsupervised model designed to work on legal cases can be employed.

This can be shown by employing the *CaseSummarizer* model [4]. The authors designed the unsupervised model to work on Australian case judgements. The NLTK (Natural Language Toolkit) library was used for standard preprocessing which included stemming, lemmatization and clearing of stop words. The TF\*IDFmatrix was used for sentence scoring built on thousand of legal case reports. These scores are summed over each sentence and normalized by the sentence length. The model used occurrences of known entities, dates, and proximity to section headings to score sentences. A new score, wnew, is calculated for the sentence using the formula; wnew = wold + σ (0.2d + 0.3e + 1.5s) where *d* is the number of ‘dates’ present in the sentence, *e* is the number of named entity mentions, *s* is a Boolean indicating the start of the section (sentences at the start of a section are given more weightage), and σ is the standard deviation among the sentence scores.

Both these techniques of summarization were “extractive summarization”. Extractive systems generate summaries by recognizing (and then concatenating) the most essential sentences in a document [5]. One of the main problems with this type of summarization is that the model considers all uses of the period symbol (‘.’) as a full stop or an end of the sentence. This results in half-cut sentences when the symbol is used to denote dates, currency units (Rs.) or designations (Dr.).

* 1. Domain independent summarization techniques

We used the Latent Semantic Analysis (LSA) summarizer available at [6]. This is also an extractive summarization technique that extracts and prepares the summary in the order in which they appear in the source document.

The first abstractive model we tried out was the T5 text-to-text transformer model from Google available at [7]. This model flushed out very small summaries and there were problems with grammar such as punctuations and articles.

* 1. Paraphrasing using Natural Language Processing (NLP)

Previous works on paraphrasing methods have considered different levels of paraphrasing granularity and different languages. While word or phrase-level paraphrasing is considered simpler, the authors in [8] used multiple sentence alignment to address sentence-level paraphrasing. Research has not been limited to the English language, as [9] presented an approach to paraphrase Hindi sentences using NLP.

1. Proposed method
   1. LexRank summarization algorithm

LexRank is an unsupervised graph-based technique for automatic text summarization. The graph method is used for sentence scoring. LexRank is used to compute sentence relevance using the notion of eigenvector centrality in a sentence graph representation [10]. This extractive algorithm works by extracting similar set of sentences with the same intent. This technique comprehends the similarity between two sentences and generates a closeness score for each pair of sentences according to which the pair is given an equivalent weight. The total score of a sentence is finalized by accounting the weight of the edges connected with it [11]. This can be visualized in Figure 1.

w12

w11

w10

w7

w2

w1

S6

S2

S1

w15

w8

w9

w6

w3

w13

w14

S3

S5

w5

w4

S4

**Fig. 1.** Graphical Representation of LexRank

The LexRank algorithm is similar to the TextRank algorithm which uses typical PageRank approach. LexRank considers position and length of the sentences while TextRank does not, making LexRank very effective for Legal documents.

* 1. PEGASUS Transformer

In NLP, transfer learning and pretrained language models have pushed the boundaries of language understanding and generation. The PEGASUS model was first proposed in [12]. The authors built upon the success of seq2seq learning techniques and unsupervised language models like ELMo and BERT to build PEGASUS. The model uses and encoder-decoder model for sequence to sequence learning. PEGASUS also adopts the state-of-the-art transformer architecture. The way it differs from previous state-of-the-art models is the pre-training. The authors pre-trained the model on a large corpus of 350 million webpages and 1.5 billion news articles to automate the selection of important sentences using the ROUGE1-F1 metric.

* 1. Implementation

The document was first summarized using LexRank model. The summary length was chosen as 35% of the original length of the document. We used the PEGASUS transformer model on a sentence level. The extractive summary generated by the LexRank model was split into sentences and fed to PEGASUS. The transformer then derived paraphrases for each sentence, making it simple to comprehend. The sentences were then merged to give a final abstractive level summary. The implementation flowchart can be visualized in Figure 2.

Input Text

Sentences

(35% of original doc)

Split

Similarity Matrix

Sentence Rankings

Summary

All Sentences

Split

Paraphrase each sentence

Final Summary

LexRank

PEGASUS

Merge

**Fig. 2.** Flowchart of implementation

1. Results
   1. ROUGE Scores

**Table 1.** Recall, Precision and F1 scores for all models implemented

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | ROUGE1 | | | ROUGEL | | |
|  | Recall | Precision | F1 | Recall | Precision | F1 |
|  |  |  |  |  |  |  |
| LSA | 0.882 | 0.536 | 0.667 | 0.882 | 0.536 | 0.667 |
| LexRank | 0.804 | **0.631** | **0.707** | 0.784 | **0.615** | **0.689** |
| CaseSummarizer | 0.451 | 0.397 | 0.422 | 0.431 | 0.379 | 0.404 |
| Graphical Model | **0.902** | 0.422 | 0.575 | **0.902** | 0.422 | 0.575 |
| T5 | 0.176 | 0.237 | 0.202 | 0.157 | 0.211 | 0.179 |

We manually summarized a document for extractive summarization evaluation and used it as a reference to calculate the ROUGE scores of all the model implemented. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation which is a metric used for determining how good the summarization model is. From Table 1, it can be observed that while the Graphical Model had the highest recall, the LexRank model had far greater precision compared to others and subsequently a higher F1 score for both ROUGE1 and ROUGEL. Thus, LexRank does the best work of extracting information which is relevant and needed then the other models.

* 1. Sample Output

We scrapped some documents from the ‘High Court of Bombay’ from [13]. Below is a sample input and output of a document dated 18th January 2021. We picked a small document for this paper, but the results stay consistent over any large document.

### Input –

*1. Heard the learned Counsel for the appellants and the*

*respondents at a considerable length.*

*2. Perused the record.*

*3. The Appeals from Order can be disposed of in following*

*terms in view of the peculiar circumstances of the case.*

*5-92265-2020-AOst-order=.doc*

*4. The operative part of the impugned order dated 20th*

*February, 2020 passed by the learned Joint Civil Judge, Senior*

*Division, Pune is defective in the sense that at the interlocutory*

*stage, the relief as has been granted is as good as a final relief in*

*the suit. As such, the said order is set aside and substituted as*

*follows :-*

*ORDER*

*(a) The appellants – original defendants shall in view of its*

*standing committee’s Resolution No.62 dated 22nd June, 2020*

*which came to be passed after the impugned order, accept the*

*supply of uniforms of the students for the Academic Year 2020-21.*

*(b) The learned Civil Judge, Senior Division, Pune shall expedite*

*the suit.*

*(c) Parties shall co-operative with the expeditious disposal of the*

*suit without seeking unnecessary adjournments.*

*5. The Appeals stand disposed of in the aforesaid terms. No*

*order as to costs.*

*6. In view of disposal of the appeals, pending Interim*

*Applications are also disposed of.*

*(PRITHVIRAJ K. CHAVAN, J.)*

Output after LexRank summarization –

*Heard the learned Counsel for the appellants and the respondents at a considerable length.*

*Perused the record.*

*The operative part of the impugned order dated 20th February, 2020 passed by the learned Joint Civil Judge, Senior Division, Pune is defective in the sense that at the interlocutory stage, the relief as has been granted is as good as a final relief in the suit.*

*As such, the said order is set aside and substituted as follows :-*

*In view of disposal of the appeals, pending Interim Applications are also disposed of.*

Output after PEGASUS paraphrasing –

*Heard the Counsel for the respondents. Used the record. The relief granted by the Joint Civil Judge, Senior Division, Pune in the impugned order dated 20th February, 2020 is not as good as the final relief in the suit. The said order is set aside and replaced. In view of the disposal of the appeals, pending interim applications are also done away with.*

It can be observed that the final summary is about a quarter in length of the original document. The LexRank algorithm extracted information which is relevant and essential, while the PEGASUS transformer paraphrased each sentence in a language which is easier to comprehend for the common public. The final output acts as an abstractive summarization which has all the relevant information from the source document and also is easy to understand.

1. Conclusion and Future Scope

Inculcating Artificial Intelligence and Machine Learning into the Judiciary System has been an area of active research in the past decade. The primary virtue of the judicial system is to pass fair and just judgements. Thus, artificial intelligence cannot entirely replace a human judge as human instincts are vital in determining an outcome of the case. AI can however be used to supplement the courts and help lawyers and judges deliver fair and speedy judgements. Automatic summarization of case documents can help the lawyers prepare for a case quickly and also help the common public truly understand case details and outcomes. In this project, we looked at various existing methods of summarization and proposed a novel method of abstractive summarization by using paraphrasing of summaries generated by an extractive summarization model. The summaries generated held all the relevant information from the source document and was also in a simpler language which makes it easy for the common public to comprehend the document. The model, however, does not deal with specific legal jargons which can be a future area of study. Apart from jargons, if the document references an existing law or rule, the summary could provide a brief passage about what the rule is. There is also a need to have automatic categorization of similar court cases and verdicts which would help the lawyers. Research of infusing AI and ML into the Judiciary System is still fledgling, and a lot of further research is possible to enhance the working of the Judiciary.

References

1. Common Law, <https://en.wikipedia.org/wiki/Common_law>, last accessed 2022/01/03
2. Precedent, <https://en.wikipedia.org/wiki/Precedent>, last accessed 2022/01/03
3. Saravanan, M., Ravindran, B., & Raman, S: Improving legal document summarization using graphical models. *Frontiers in Artificial Intelligence and Applications*, *152*, 51. (2006)
4. Polsley, S., Jhunjhunwala, P. and Huang, R.: 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. (2016)
5. Narayan, S., Cohen, S. B., & Lapata, M. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636,* (2018).
6. Sumy 0.9.0, <https://pypi.org/project/sumy/>, last accessed 2021/12/26
7. Simple abstractive text summarization with pretrained T5 — Text-To-Text Transfer Transformer, <https://towardsdatascience.com/simple-abstractive-text-summarization-with-pretrained-t5-text-to-text-transfer-transformer-10f6d602c426>, last accessed 2021/12/26
8. Barzilay, R., & Lee, L. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *arXiv preprint cs/0304006*, (2003)
9. Sethi, N., Agrawal, P., Madaan, V., & Singh, S. K. A novel approach to paraphrase Hindi sentences using natural language processing. *Indian Journal of Science and Technology*, *9*(28), 1-6 (2016).
10. LexRank method for Text Summarization, <https://iq.opengenus.org/lexrank-text-summarization/>, last accessed 2021/12/27
11. Samuel, A., & Sharma, D. K. Modified lexrank for tweet summarization. *International Journal of Rough Sets and Data Analysis (IJRSDA)*, *3*(4), 79-90 (2016).
12. Zhang, J., Zhao, Y., Saleh, M. and Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: *International Conference on Machine Learning*, pp. 11328-11339. PMLR (2020).
13. High Court of Bombay, <https://bombayhighcourt.nic.in/index.php>, last accessed 2022/01/02