Legal Knowledge and Information Systems
E. Schweighofer (Ed.)

doi:10.3233/FAIA210320

Improving Legal Case Summarization Using Document-Specific Catchphrases

Arpan Mandal[a], Paheli Bhattacharya[b], Sekhar Mandal[a], Saptarshi Ghosh[b]
[a] *Indian Institute of Engineering Science and Technology Shibpur, India*
[b] *Indian Institute of Technology, Kharagpur, India*

Abstract. Legal case summarization is an important problem, and several domain-specific summarization algorithms have been applied for this task. These algorithms generally use domain-specific legal dictionaries to estimate the importance of sentences. However, none of the popular summarization algorithms use *document-specific catchphrases*, which provide a unique amalgamation of domain-specific and document-specific information. In this work, we assess the performance of two legal document summarization algorithms, when two different types of catchphrases are incorporated in the summarization process. Our experiments confirm that both the summarization algorithms show improvement across all performance metrics, with the incorporation of document-specific catchphrases.

Keywords. Legal case document, Summarization, Catchphrases

1. Introduction

Summarization of legal case documents is an important problem and has been well-studied by researchers [1–4]. Summarization algorithms can be of different types, viz. *extractive* vs. *abstractive*, *unsupervised* vs. *supervised*. Most summarization algorithms developed for the legal domain are extractive and unsupervised in nature, mainly due to the lack of large training data in the legal domain.

These algorithms being extractive in nature, attempt to assign a likelihood-score to each sentence of a document, and choose the top-scoring sentences as the summary of the document. While measuring the score of a sentence (the likelihood of the sentence to be included in the summary), various summarization methods consider either or both of two factors – (1) *document-specific importance* of the sentence with respect to other sentences in the document, (2) *domain-specific importance* of the sentence, e.g., several legal domain-specific algorithms use an external set of legal terms (a legal dictionary) and consider the number of legal terms contained in a sentence [2, 3].

Although these two factors are independently used to characterize the likelihood-score of a sentence, we hypothesize – *"an appropriate amalgamation of document-specific and domain-specific importance may provide new useful information to the summarization algorithms, which can subsequently improve* *their performance"*. This combined information can be provided by the use of ***document-specific catchphrases*** that are a set of short (one-word or multi-word) phrases that collectively provide a concise representation of a legal document [5–8]. These catchphrases are not only legal domain-specific important terms, but also terms or phrases that have document-specific importance. Although domain-specific dictionaries have widely been used in summarization algorithms [2,3], catchphrases are different from domain-specific dictionaries in that they also capture document-specific important terms (which may not be legal keywords).

In this work, we investigate whether using document-specific catchphrases can improve the performance of legal document summarizers. To this end, we use two different types of catchphrases – extracted from a case document by two methods *PSLegal* [9] and *D2V-BiGRU-CRF* [10] (details in Section 2) – to aid two legal-specific summarization techniques – *DELSumm* [3] and, *CaseSummarizer* [2] (details in Section 3). We conduct experiments over a set of case documents from the Indian Supreme Court (details in Section 4). Our experiments demonstrate that, the performances of both the summarization algorithms improve when document-specific catchphrases are incorporated.

2. Related Works

In this work, we propose to use document-specific catchphrases to improve legal case document summarization. In this section, we survey some catchphrase detection methods and case document summarization methods.

2.1. Legal catchphrase extraction

Several catchphrase detection methods have been developed for legal documents [9–11]. We briefly discuss two catchphrase extraction methods developed in our prior works, both of which provide meaningful catchphrases that agree with those chosen by law domain experts [9, 10].

PSLegal [9] – an unsupervised method: Given the text of a document d, this method involves two major steps to extract the set of catchphrases:
Step 1: Some *candidate phrases* are extracted from d. These are actually *noun phrases* extracted using a customized set of grammatical rules (details in [9]).
Step 2: Next, an appropriate scoring function takes as input the text of d and a candidate phrase c, and computes the likelihood for c to be a catchphrase for the document d. The scoring function takes into account three factors – (1) document-specificity of the phrase c, (2) domain-specific importance of c, (3) presence of a predefined legal term within the phrase c. The final PSLegal score is the product of these three factors. All candidate phrases are scored using this scoring function and then 10% of the top-scored ones are chosen as the catchphrases for the given document. Further details of the method can be found in [9], and a ready-to-use implementation of this algorithm is available at `https://github.com/amarnamarpan/PSLEGAL`.

D2V-BiGRU-CRF [10] – a supervised method: This is a *neural sequence tagging model* that has the ability to be trained over a relatively small training dataset (typically, a few hundred documents and their gold standard catchphrases) and then the trained model can be used to extract catchphrases from unseen case documents. It takes as input a sequence of words and identifies each word to be either a part of a catchphrase or not.

The D2V-BiGRU-CRF architecture (details in [10]) employs these layers – (1) a bidirectional language model [12] that extracts word and character embeddings. (2) a BiGRU layer that combines both the embeddings, (3) a fully connected layer that learns a representation of the outputs of the BiGRU layer and a Doc2vec [13] embedding of the input document, and (4) a Conditional Random Field (CRF) layer which predicts whether a word is part of a catchphrase or not. It was demonstrated in [10] that the D2V-BiGRU-CRF extracts catchphrases that match well with those selected by law domain experts. An implementation of this model along with our trained model is available at `https://github.com/amarnamarpan/D2V-BiGRU-CRF`.

2.2. Summarization of court case documents

Many general (domain-independent) text summarization algorithms have been used for summarization of legal case documents, e.g., in [3,14]. Additionally, many algorithms have been developed specifically for the summarization of legal case documents, such as LetSum [15], K-mixture model [4], CaseSummarizer [2] and DELSumm [3]. Out of these, in this work, we use document-specific catchphrases to improve the performances of CaseSummarizer [2] and DELSumm [3].

3. Incorporating catchphrases in legal case summarization algorithms

In this section, we describe two unsupervised, extractive summarization algorithms built for summarizing legal case documents – (1) DELSumm [3] and, (2) CaseSummarizer [2] – and how each of these algorithms can be modified to incorporate document-specific catchphrases.

3.1. **DELSumm**

DELSumm [3] is a recently developed algorithm that models the problem of summarizing a case document as maximizing an Integer Linear Programming (ILP) objective function that maximizes the inclusion of the most informative sentences in the summary. As input, DELSumm takes - (1) a case document where each sentence is marked with its rhetorical role, out of the eight roles (e.g., Facts, Arguments, Ratio of the decision, Ruling), and (2) the desired length L of the summary in words. The output is a summary whose length is at most L, and contains sentences from each of the rhetorical roles. The algorithm considers a set of guidelines suggested by law experts as to how the different rhetorical segments of a case document should be summarized. To judge the informativeness of a sentence, the algorithm considers, among other factors, a set of *content words* which are basically terms in a legal dictionary compiled from various sources. More details can be found in [3].

Incorporating catchphrases in DELSumm: We replace the legal content words (described above) by a set of catchphrases extracted from the input document (that is to be summarized). In the original DELSumm, the set of content words remains the same for every document. Whereas, now the algorithm gets modified in a way whereby, while constructing the summary, it gives more importance to the sentences that contain catchphrases specific to each input document.

3.2. CaseSummarizer

CaseSummarizer (CaseSumm in short) [2] uses a specialized scoring function to score each sentence in a case document, and then chooses the highest scored sentences to build the summary. To build the scoring function for sentences, it considers three factors – (1) *the occurrence of known important entities in a sentence (the entities were marked by domain experts in the original work [2])*, (2) the occurrence of dates in a sentence, and (3) the proximity of a sentence to section headings. First, an initial score/weight is computed for each sentence by summing the TF-IDF scores of the constituent words and normalizing over the the sentence length; this score is called w_{old}. A new score w_{new} is then computed for a sentence as $w_{new} = w_{old} + \sigma(0.2d + 0.3e + 1.5s)$ where, d is the number of dates in the sentence, e refers to the number of entities, s is a boolean variable specifying whether the sentence is the first sentence in a section, σ is the standard deviation between the scores of all sentences, and the coefficients are selected through trial-and-error and feedback from experts.

Incorporating catchphrases in CaseSumm: We modify CaseSumm by incorporating document-specific catchphrases (e.g., those generated by D2V-BiGRU-CRF or PSLegal) in place of the *entities* in the document. More specifically, in the expression for w_{new} stated above, we replace the term e (which signifies the number of entities in a sentence, in the original algorithm) with the number of document-specific catchphrases contained in the sentence.

4. Dataset and Experimental Results

Dataset for evaluation of summarization performance: We reuse the dataset and evaluation setup from our recent work [3]. The dataset consists of 50 legal case documents from the Indian Supreme Court, along with their summaries (of length approximately one-third of the original document lengths) written by two domain experts (senior Law students from the Rajiv Gandhi School of Intellectual Property law, one of the most reputed Law schools in India).[1] As document summarization is a subjective task, the two experts wrote two separate summaries; all scores presented in the paper are averaged over the two gold standard summaries written by the two experts.

Metrics for summarization performance: We compare the match between the algorithmic summaries and the gold standard summaries, and report Rouge-2 (considers bigram matches) and Rouge-L (considers Longest Common Subsequence matches) Recall and F-scores. All scores for a particular document are averaged

[1] Also, each sentence in the documents is labeled with its rhetorical role by the same experts; these labels are used by DELSumm (details in [3]).

Table 1. Comparing the performance of original DELSumm (abbreviated as DLS) and DELSumm with different variations of catchphrases. DBC: catchphrases extracted by D2V-BiGRU-CRF; PSL: catchphrases extracted by PSLegal; Ldict: the Legal Dictionary used by the original summarization algorithm. The highest value for each metric is in bold-fonts.

	Rouge-2 R	Rouge-2 F	Rouge-L R	Rouge-L F
Original DELSumm (DLS)	0.4323	0.4217	0.6831	0.6017
DLS with DBC	0.4588	0.4411	0.6892	0.6102
DLS with DBC & Ldict	0.4557	0.4372	**0.6909**	0.6096
DLS with PSL	**0.4593**	**0.4435**	0.6763	0.6111
DLS with PSL & Ldict	0.4479	0.4343	0.6805	0.6105
DLS with PSL & DBC	0.4574	0.4422	0.6757	0.6103
DLS with PSL, DBC & Ldict	0.4509	0.4365	0.6828	**0.6118**

Table 2. Comparing the performance of original CaseSumm (CSM) and CaseSumm with different variations of catchphrases. The highest value for each metric is in bold-fonts. The terms DBC, PSL and Ldict are as explained in the caption of Table 1

	Rouge-2 R	Rouge-2 F	Rouge-L R	Rouge-L F
Original CaseSumm (CSM)	0.3198	0.3636	0.5415	0.5343
CSM with DBC	0.3258	0.3726	0.5490	**0.5426**
CSM with DBC & Ldict	**0.3265**	**0.3738**	**0.5493**	0.5425
CSM with PSL	0.3221	0.3690	0.5465	0.5397
CSM with PSL & Ldict	0.3221	0.3690	0.5465	0.5397
CSM with PSL & DBC	0.3220	0.3689	0.5463	0.5396
CSM with PSL & DBC & Ldict	0.3229	0.3702	0.5473	0.5411

over the two gold standard summaries written by the two experts for the document (as stated above).

Variations of the summarization methods tried: As was explained in Section 3, we shall compare the performance of a summarization algorithm (DELSumm / CaseSumm) when used in its original setting (with an in-built legal dictionary, referred to as 'LegDict'), and when used with document-specific catchphrases. We have two kinds of catchphrases – (1) those generated by PSLegal (referred to as PSL), and (2) those generated by D2V-BiGRU-CRF (referred to as DBC). Thus, for a summarization algorithm (say, S) that originally uses the legal dictionary 'LegDict', we experiment with the following variations: (1) Original S with the default LegDict, (2) S with only DBC, (3) S with DBC and LegDict, (4) S with only PSL, (5) S with PSL and LegDict, (6) S with both DBC and PSL, (7) S with all three – both type of catchphrases, and the LegDict.

In variations (2) and (4) stated above, the catchphrases identified by D2V-BiGRU and PSLegal respectively are used *in place of* the default LegDict in the original summarization algorithms. Whereas, in the variations (3), (5), and (7), the catchphrases identified by D2V-BiGRU and/or PSLegal are used in conjunction with LegDict.

Results: Table 1 shows the performance of the original DELSumm and DELSumm with different variations of catchphrases (as explained above). We see that, in all cases, use of document-specific catchphrases leads to better summarization than what is achieved by the original DELSumm. The best performance is achieved when DELSumm is used along with catchphrases identified by PSLegal – in this setting, the Rouge-2 F-score increases from 0.4217 (for the original DEL-

Summ) to 0.4435. Note that the original DELSumm already out-performs several other summarization methods on this dataset, as was shown in [3]. Incorporating document-specific catchphrases improves the summmarization even further.

Similarly, Table 2 shows the performance of the original CaseSummarizer and its variations with different sets of catchphrases. In all cases, use of document-specific catchphrases leads to better summarization than what is achieved by the original CaseSummarizer.

5. Conclusion

We show that using document-specific catchphrases can improve the performance of existing summarization algorithms while summarizing legal case documents.

While the present work considers only unsupervised and extractive summarization algorithms, in future, we plan to explore ways of improving *supervised* and *abstractive* summarization algorithms using catchphrases.

References

[1] Erkan G, Radev DR. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. J Artif Int Res. 2004;22(1):457–479.

[2] Polsley S, Jhunjhunwala P, Huang R. CaseSummarizer: A System for Automated Summarization of Legal Texts. In: Proceedings of COLING 2016; 2016. .

[3] Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S. Incorporating Domain Knowledge for Extractive Summarization of Legal Case Documents. In: Proc. ICAIL; 2021. .

[4] Saravanan M, Ravindran B, Raman S. Improving Legal Document Summarization Using Graphical Models. In: Proc. of the Conference on Legal Knowledge and Information Systems: JURIX 2006. IOS Press; 2006. p. 51–60.

[5] Galgani F, et al. Towards Automatic Generation of Catchphrases for Legal Case Reports. In: Proceedings of Computational Linguistics and Intelligent Text Processing; 2012. .

[6] Witten IH, et al. KEA: Practical Automatic Keyphrase Extraction. In: Proceedings of the Conference on Digital Libraries; 1999. p. 254–255.

[7] Liu Z, et al. Clustering to Find Exemplar Terms for Keyphrase Extraction. In: Proc. of the Conference on Empirical Methods in Natural Language Processing; 2009. p. 257–266.

[8] Wu YFB, Li Q. Document Keyphrases as Subject Metadata: Incorporating Document Key Concepts in Search Results. Information Retrieval Journal. 2008;11:229–249.

[9] Mandal A, Ghosh K, Pal A, Ghosh S. Automatic Catchphrase Identification from Legal Court Case Documents. In: Proc. ACM Conference on Information and Knowledge Management (CIKM); 2017. .

[10] Mandal A, Ghosh K, Ghosh S, Mandal S. A sequence labeling model for catchphrase identification from legal case documents. Artificial Intelligence and Law. 2021 June.

[11] Tran VD, Nguyen ML, Satoh K. Automatic Catchphrase Extraction from Legal Case Documents via Scoring using Deep Neural Networks. CoRR. 2018;abs/1809.05219.

[12] Peters M, et al. Deep Contextualized Word Representations. In: Proceedings of the North American Chapter of the Association for Computational Linguistics; 2018. p. 2227-37.

[13] Le Q, Mikolov T. Distributed Representations of Sentences and Documents. In: Proceedings of International Conference on Machine Learning; 2014. p. 1188-96.

[14] Bhattacharya P, Hiware K, Rajgaria S, Pochhi N, Ghosh K, Ghosh S. A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments. In: Advances in Information Retrieval; 2019. p. 413-28.

[15] Farzindar A, Lapalme G. LetSum, an automatic Legal Text Summarizing system; 2004. .