Automatic Catchphrase Identification from Legal Court Case Documents

Arpan Mandal
IIEST Shibpur, India

Kripabandhu Ghosh
IIT Kanpur, India

Arindam Pal
TCS Research, India

Saptarshi Ghosh
IIT Kharagpur, India; IIEST Shibpur, India

ABSTRACT

Automatically identifying catchphrases from legal court case documents is an important problem in Legal Information Retrieval, which has not been extensively studied. In this work, we propose an unsupervised approach for extraction and ranking of catchphrases from court case documents, by focusing on noun phrases. Using a dataset of gold standard catchphrases created by legal experts from real-life court documents, we compare the proposed approach with several unsupervised and supervised baselines. We show that the proposed methodology achieves statistically significantly better performance compared to all the baselines.

CCS CONCEPTS

•**Information systems** →**Information retrieval;**

KEYWORDS

Legal IR; Catchphrase extraction; Court cases

1 INTRODUCTION

In countries following the *Common Law system*, in an ongoing legal case, great emphasis is given on prior cases / precedences, on the principle that similar facts must not be differently interpreted on repeated occurrence.[1] In such legal systems, this reliance on precedences makes it critical for legal practitioners to find and study older cases, in order to analyze how the issues related to the current case were discussed and ruled in preceding cases.

The ever-increasing volume of legal text makes information access from them challenging for legal practitioners. Court case descriptions are usually long and contain complex sentences [1], which makes thorough reading strenuous and time-consuming. So, it is essential for legal practitioners to have systems which can automatically provide a gist of the legal concepts in the whole case.

To this end, identification of *catchphrases* or *catchwords* [4] from legal documents is of particular interest (the terms *catchphrases* and *catchwords* will be used interchangeably in this paper).[2] Catchphrases can crucially help the lawyers in quickly understanding the underlying legal concepts pertaining to a case. Till now, catchphrase identification from legal documents is mostly done manually by legal experts. For instance, the *Manupatra legal search system* (http://www.manupatra.co.in/), which is popular among legal practitioners in India, provides manually extracted catchwords for the cases in the important law courts in India. However, as the number of legal documents available online is increasing, it is becoming difficult for legal experts to manually extract catchwords from the huge number of documents. In such situations, methodologies for automated extraction of catchwords are essential.

Challenges: Automatic identification of legal catchphrases is a challenging problem. To give an idea of the challenging nature of the problem, Table 1 shows three example cases from the Manupatra online legal search system, showing their unique case ids and some catchphrases chosen by legal experts. For example *1953.INSC.24* is the *Case id* of the case titled "THE STATE OF BOMBAY & ANR V. THE UNITED MOTORS (INDIA) LTD. & ORS" (URL given in the Table). For this case, some example catchphrases are 'Appropriate', 'Assessment', etc. Catchphrases *occur in low numbers* in court case documents and some of them can occur in only a few documents in a large collection. They are of *variable lengths*, typically as *n*-grams where *n* can be as large as 7. Additionally, the presence of stopwords in catchphrases like 'Directive Principle of State Policy', 'Adventure in the Nature of Trade' etc., prohibits the conventional removal of stopwords as a pre-processing step prior to retrieval. All these characteristics of legal catchphrases hint at exploration beyond traditional IR techniques.

Prior work: To our knowledge, there is only one prior work which attempted to automatically extract catchwords from legal documents. Galgani *et al.* [3] reported a work on identification of *sentences containing catchphrases* and evaluation of the underlying mechanism. However, their goal was to summarize the documents, and for that, it was sufficient to extract the sentences containing the catchwords. We focus on a different problem – *automatic identification of the exact catchphrases* from a given court case. Additionally, we show that our proposed approach achieves better extraction of legal catchphrases than the method in [3], applied specifically for identifying catchphrases.

Present work: In this paper, we propose an automated methodology for extraction of catchwords from legal documents. We also

[1] https://en.wikipedia.org/wiki/Common_law as seen on 23rd May, 2017.

[2] https://www.irwinlaw.com/cold/catchwords

CIKM'17 , November 6–10, 2017, Singapore

978-1-4503-4918-5/17/11...$15.00
DOI: http://dx.doi.org/10.1145/3132847.3133102

Case Id	Catchphrases
1953.INSC.24 http://liiofindia.org/in/cases/cen/INSC/1953/24.html	Actual Delivery, Advocate-General, Alternative Remedy, Appropriate, Assessment, Carrier, Carrying on Business, Cause of Action, Commencement of the Constitution, Competent Legislature, Consignment, Constitution of India, Constitutional Validity, Consumption, Contract, Contract of Sale, Contravention, Cost, Dealer, Declared by Parliament, Deduction, Definition, Delegate, Demand Notice, Despatch, Discrimination, Discriminatory, Double Taxation, Existing Law, Export, Federal Court, Freedom of Trade
1991.INSC.12 http://liiofindia.org/in/cases/cen/INSC/1991/12.html	Advisory Board, Allowance, Appropriate Government, Arrest, Constitutional Obligation, Constitutional Question, Constitutional Safeguard, Detaining Authority, Detention, Detenu, Duty of the State, Earliest Opportunity, General Clauses Act, Grounds of Detention, Guarantee, Legal Obligation, Liberty, Order of Detention
1983.INSC.27 http://liiofindia.org/in/cases/cen/INSC/1983/37.html	Commutation, Confinement, Conspiracy, Constitution of India, Death Sentence, Fundamental Right, Imposition of Death Sentence, Judicial Proceeding, Life Imprisonment, Solitary Confinement, Speedy Trial, Transportation for Life

Table 1: Examples of Indian Supreme Court cases and catchphrases taken from the Manupatra legal expert system

propose a novel scoring function for ranking the extracted catchphrases. We perform experiments over a dataset of 400 case documents from the Supreme Court of India cases (archived at http://liiofindia.org/in/cases/cen/INSC/), and gold standard catchwords identified manually by legal experts in the Manupatra legal search system.

We compare our approach with a *supervised* keyphrase extraction tool, Kea [9] and several unsupervised term-weighting baselines. We show that our proposed unsupervised methodologies statistically significantly outperform all the standard baselines.

2 METHODOLOGIES FOR CATCHPHRASE EXTRACTION

The task of automatic catchphrase extraction from a given set of documents can be decomposed into the two following sub-tasks:
(1) Selection of candidate catchphrases: In this step, for a given document, we look to identify parts of the document text which can act as potential catchphrases.
(2) Scoring and ranking of candidate catchphrases: In this step, we assign scores to the candidate phrases and rank them on the basis of these scores. We propose a novel algorithm to score the candidate phrases.
We now describe methodologies for automatic extraction of catchphrases from legal documents (specifically, court case descriptions from the Supreme Court of India).

2.1 Selection of candidate catchphrases

Through manual inspection, we conjectured that most of the catchphrases in legal documents are noun phrases. To strengthen our conjecture, we consulted some legal experts, who confirmed that many of the important catchphrases in legal case documents are noun phrases. For empirical verification, we viewed catchphrase extraction task as a pattern matching problem and experimented with various types of patterns that are likely to match legal catchphrases. We extracted (i) word n-grams ($n = 1, 2, \ldots, 7$), and (ii) noun phrases from legal texts, and checked which type of patterns match better with catchphrases that are actually identified by legal experts (using our dataset described in the introductory section). We achieved a much higher recall in identifying catchphrases by using noun phrases as candidate phrases.

Hence, noun phrases were extracted using a standard noun phrase chunker.[3] Extraction of noun phrases requires a *grammar* that defines the structure of a noun phrase in terms of the parts-of-speech (POS) tags of the constituent words. The structure of a noun phrase in English can be quite complex [2]. For the purpose of extracting legal catchphrases, we used a simplified version of the grammar in [2], involving only adjectives, nouns and prepositions.

The patterns extracted from a document can be in singular or plural senses, while legal catchphrases are usually in singular (e.g., 'Fundamental Right' and 'Legal Obligation', and *not* 'Fundamental Rights' or 'Legal Obligations'). Hence, we adopt the following approach. We first use a Parts-of-Speech (POS) tagger to identify the nouns in the extracted patterns, and then use *lemmatization* on the nouns, to remove the inflectional endings. Note that we perform lemmatization only on the nouns, and *not* on other words, since inflectional endings in adjectives are often parts of legal catchphrases (e.g., 'Constitutional Validity', 'Discriminatory Evidence').

2.2 Scoring of the selected candidate phrases

After the extraction of candidate phrases from a particular document d, we rank them based on a scoring technique. We describe below some standard scoring methods, which we consider as baselines. We also present our proposed scoring scheme thereafter. Most of these schemes are designed on TF-IDF based scoring which has been conventionally used for scoring query terms; here we use it for scoring noun phrases instead. We use a number of notations in the rest of this section, that are defined in Table 2.

(1) **BM25 :** The BM25 function is well-known for scoring documents with respect to a given query [5]. Here we use this function for assigning score to an extracted candidate phrase c in a given document d. We use the following scoring function to assign score to our candidate phrases:

$$Score(c,d) = IDF(c) \cdot \frac{TF(c,d) \cdot (k_1 + 1)}{TF(c,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \quad (1)$$

[3] We used the Python Natural Language Toolkit for noun phrase chunking (http://www.nltk.org/howto/chunk.html) as well as for subsequent steps like POS tagging and lemmatization.

Notation	Meaning
c	a multi-word candidate phrase
$\|c\|$	number of words in a candidate phrase
t	A term or a single word
t_i	the i^{th} term in a multi-word candidate phrase
d	a single document
$\|d\|$	length of a document in words
$\mathbb{C}$	collection of documents of any domain
$\mathbb{C}_l$	collection of documents of the legal domain
$\mathbb{C}_{nl}$	collection of documents of a non-legal domain
$\mathbb{D}$	manually constructed legal dictionary
n	total number of documents in the corpus
$avgdl$	average of all document lengths in the corpus

Table 2: Notations used for describing the scoring methods.

where $TF(c,d)$ is the frequency of the candidate phrase c in document d. k_1 and b are free parameters whose values we choose as $k_1 \in [1.2, 2.0]$ and $b = 0.75$.[4] $IDF(c)$ is the inverse document frequency of the candidate phrase c, calculated as:

$$IDF(c) = \log \frac{n - DF(c) + 0.5}{DF(c) + 0.5} \tag{2}$$

where $DF(c)$ (Document Frequency) is the number of documents in which the the phrase c occurs in the collection.

(2) **KLIP:** This is a KL-divergence based score [7, 8], where the final score is a sum of two different scores – (1) KL informativeness (KLI), and (2) KL phraseness (KLP).

The KLI score measures how well does a candidate phrase c represent a document d. It is calculated as:

$$KLI(c,d) = \frac{TF(c,d)}{|d|} \cdot \log \frac{(\frac{TF(c,d)}{|d|})}{(\frac{CF(c)}{n})} \tag{3}$$

where, $CF(c)$ is the total number of occurrences of c in the whole corpus.

On the other hand, the KLP score is meant specifically for multi-word phrases. It compensates for the low frequency of multi-word phrases by assigning higher weights to longer phrases:

$$KLP(c,d) = \frac{TF(c,d)}{|d|} \cdot \log \frac{(\frac{TF(c,d)}{|d|})}{\prod_{i=1}^{|c|}(\frac{Freq(t_i,d)}{|d|})} \tag{4}$$

where t_i is the i^{th} unigram of the phrase c, and $Freq(t_i,d)$ is the frequency of the unigram t_i in document d.

(3) **MyScore**: This score is generated by an unsupervised method described in [3] for ranking sentences containing catchphrases in legal documents. For our purpose, rather than scoring sentences we scored the candidate phrases. *MyScore* is a combination of three scores — TF-IDF, TF, and a new score named *NOccur*. In this new metric *NOccur*, firstly the important sentences are marked within a document. Hence, the words that appear in these important sentences are weighted higher than the remaining words. We consider this baseline since, to the best of our knowledge, this is the only previous work adjacent to our problem in the legal domain.

[4]https://en.wikipedia.org/wiki/Okapi_BM25 as seen on 23rd May, 2017.

(4) **Proposed:** We propose a novel technique which aims at estimating the importance of a term specifically in the legal domain relative to a non-legal domain. To this end, we represent the non-legal domain using a collection of documents from the famous *20 Newsgroups dataset* that comprises of around 18,000 newsgroup posts on 20 different topics.[5] Now, for each term we try to find out its importance in both the legal and non-legal domains. We measure the importance of term t in a domain $\mathbb{C}$ as follows:

$$Importance(t,\mathbb{C}) = \frac{CF(t,\mathbb{C})}{DF(t,\mathbb{C}) + 1} \tag{5}$$

where, $CF(t,\mathbb{C})$ is the collection frequency of the term t in corpus $\mathbb{C}$, and $DF(t,\mathbb{C})$ is the number of documents in $\mathbb{C}$ that the term t occurs in.

We wish to have a scoring where terms that are more important in legal domain and less important in non-legal domain, get a higher score. Hence, we use the following scoring function:

$$Score(t,\mathbb{C}_l,\mathbb{C}_{nl}) = \frac{Importance(t,\mathbb{C}_l)}{Importance(t,\mathbb{C}_{nl}) + 1} \tag{6}$$

where $\mathbb{C}_l$ and $\mathbb{C}_{nl}$ are the legal and non-legal corpus respectively. It is to be noted that, whenever a term from $\mathbb{C}_l$ is not present in $\mathbb{C}_{nl}$, its importance in $\mathbb{C}_{nl}$ equates to zero.

Now that our terms are assigned scores, we need to finally compute the scores of our candidate phrases. To do so, we average the above score for all terms in the phrase and take a *natural logarithm* of it. Then, we multiply the KLI score of the phrase with this sum. The final score, denoted by PS, is defined as:

$$PS(c,\mathbb{C}_l,\mathbb{C}_{nl}) = \log[\sum_{i=1}^{|c|} Score(t_i,\mathbb{C}_l,\mathbb{C}_{nl})] \cdot KLI(c,d) \tag{7}$$

where t_i is the i^{th} term of the candidate phrase c. So, PS is the product of two terms. One of them is the KLI score calculated just as in KLIP method. The KLI term gives the specificity of a term in a given document, whereas the other term gives the specificity of the term in the legal domain.

Explicitly considering legal terms in scoring: We also consider a variant of our proposed method, by adding more importance to the candidate phrases that contains a legal phrase. We compiled a large *dictionary of legal terms*, obtained from several online sources.[6] The set of legal terms mostly contains unigrams, bigrams, and trigrams, though there also n-grams with higher values of n. We denote this legal dictionary by $\mathbb{D}$. We define a legal term-based score, *PSLegal* as below:

$$PSLegal(c,\mathbb{C}_l,\mathbb{C}_{nl}) = PS(c,\mathbb{C}_l,\mathbb{C}_{nl})(1 + \frac{|MAXMatch_{p\in\mathbb{D}}(p,c)|}{|c|}) \tag{8}$$

where $MAXMatch$ denotes the maximum match the phrase c has with any phrase p in the legal dictionary $\mathbb{D}$. Note that, we consider only those phrases p in $\mathbb{D}$ which are subsets of the phrase c. That is,

[5]The 20 Newsgroups dataset can be obtained from https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

[6]Legal terms were compiled from http://www.plainenglish.co.uk/files/legalguide.pdf, http://www.legalindia.com/legal-word-or-phrase/, http://dakshindia.org/common-legal-terms/.

we are looking to reward those instances in this scoring function, where a candidate phrase completely matches a legal phrase, or contains a legal phrase. We ignore the cases where a candidate phrase is a subset of a legal phrase, since this may wrongly upweight a non-legal phrase. For example, the candidate term *India* is contained in the legal phrase *Constitution of India* but *India* is not necessarily a legal term.

2.3 Supervised keyphrase extraction tool: KEA

KEA [9] is a state-of-the-art, machine learning based *supervised* baseline which extracts phrases and ranks them.[7] KEA needs some training examples of documents with annotated keyphrases. From these training examples, it learns a model for extraction of keyphrases, which can then be applied on new documents. Details of the working of KEA can be found in [9].

We used KEA to extract catchphrases from the 400 legal documents that we have. For training KEA, we used 360 documents, along with their Manupatra catchphrases extracted from these documents, as the training samples, and then used the learnt model to extract keyphrases from the other 40 documents. This process was repeated ten times, using 10-fold cross validation. Thus, KEA was used to extract keyphrases from each document once. The performance of KEA is reported in the next section.

3 EXPERIMENTS AND RESULTS

In this section, we present the results of the proposed method in comparison with the baseline methods.

3.1 Experimental setup

As stated in the introductory section, we have the catchphrases identified by the Manupatra legal expert system for a set of 400 Indian Supreme Court cases. From each document, we do the following. (i) We use our catchphrase extraction method described in Section 2.1 to obtain candidate catchphrases. Then, we rank the extracted catchphrases using our proposed ranking scheme described in Section 2.2 as well as the baseline ranking methods. (ii) We apply the supervised KEA tool which ranks the extracted phrases [9]. Then we check to what extent Manupatra catchphrases are correctly retrieved by these methods.

Evaluation measures: From a particular legal document, we obtain a ranked list of extracted patterns, where the patterns are ranked based on the various scores described in Section 2. The KEA tool also gives a ranked list of the extracted keyphrases. We evaluate the performance of a particular methodology by computing the *Mean Average precision (MAP)* over the ranked list. For a particular type of patterns, we report the mean of the Average Precision for each document over the set of 400 documents. This measure accounts for both Precision and Recall. We also report *Precision@10*, *Recall@100* and *R-Precision* averaged over all the 400 documents.

3.2 Results

Table 3 shows the performance of the different types of patterns, averaged over the 400 legal documents. The proposed methods (*PS* and *PSLegal*) statistically significantly outperform all the baselines in all the evaluation measures at 95% confidence interval ($p-value < 0.05$) by Wilcoxon signed-rank test [6] .

Method	Prec@10	Recall@100	R-Precision	MAP
BM25	0.1345	0.3978	0.1218	0.1445
KLIP	0.1382	0.3816	0.1147	0.1471
KEA	0.1008	0.1861	0.0685	0.1043
MyScore	0.0088	0.1061	0.0136	0.0300
PS	0.1598^{BKEM}	0.4121^{BKEM}	0.1351^{BKEM}	0.1752^{BKEM}
PSLegal	$\mathbf{0.1653}^{BKEM}$	$\mathbf{0.4153}^{BKEM}$	$\mathbf{0.1377}^{BKEM}$	$\mathbf{0.1770}^{BKEM}$

Table 3: Performance of automatic catchphrase extraction using different scoring methods. All measures are averaged over 400 court case descriptions from the Indian Supreme Court cases. Bold font shows the best value. The superscripts show significant improvements – B, K, E, and M indicate that the proposed method is statistically significantly better at 95% confidence interval ($p < 0.05$) than BM25, KLIP, KEA and MyScore respectively.

When compared with each other, *PSLegal* numerically outperforms *PS*, but the difference is statistically significant ($p - value < 0.05$) only for Precision@10. Hence, incorporation of a legal dictionary helps in retrieval of catchphrases, though the improvement is not always substantial. The fact that the improvement is small is primarily because a large fraction of catchwords are *not* legal terms (e.g., 'cost', 'dealer', as shown in Table 1).

4 CONCLUSION

In this work, we developed unsupervised schemes for catchphrase identification from legal documents. We introduce a new scoring function to find the importance of a term in the legal domain, in contrast to non-legal domains. Our proposed methodologies outperform several standard baselines.

In future, we plan to experiment with supervised schemes for catchword extraction. Additionally, methodologies for automatically extracting catchwords from legal documents have various potential applications, such as indexing large legal databases, retrieving relevant prior cases, summarizing legal documents [3], and so on. We look forward to exploring these directions in future.

REFERENCES

[1] Stefanie Brüninghaus and Kevin D. Ashley. 2001. Improving the Representation of Legal Case Texts with Information Extraction Methods. In *Proc. Int'l Conf. on Artificial Intelligence and Law (ICAIL)*.

[2] eng-noun-phrase 2017. Grammatical Forms of English Noun Phrases. (2017). http://www.linguisticsgirl.com/grammatical-forms-of-english-noun-phrases/.

[3] Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. Towards Automatic Generation of Catchphrases for Legal Case Reports. In *Proc. Int'l Conf. on Computational Linguistics and Intelligent Text Processing (CICLing)*.

[4] J.L.T. Olsson. 1999. Guide To Uniform Production of Judgments, 2nd edn. *Australian Institute of Judicial Administration, Carlton South* (1999).

[5] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[6] S. Siegel. 1956. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill.

[7] Takashi Tomokiyo and Matthew Hurst. 2003. A Language Model Approach to Keyphrase Extraction. In *Proc. ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment.*

[8] Suzan Verberne, Maya Sappelli, Djoerd Hiemstra, and Wessel Kraaij. 2016. Evaluation and analysis of term scoring methods for term extraction. *Information Retrieval Journal* 19, 5 (2016).

[9] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In *Proc. ACM Conference on Digital Libraries.*

[7] The software can be downloaded from http://www.nzdl.org/Kea/.