Concept Extraction from Legal Documents

Report prepared in completion of the project assigned to

Avi Chawla
Junior Undergraduate
Dept. of Computer Science and Engineering
Indian Institute of Technology (BHU)
Varanasi

Under the guidance of

Mr. Swapnil Kura, Mr. Ravindra Guntur and Mrs. Manjusha Madabushi

Talentica Software Pvt. Ltd.
Pune, Maharashtra 411045, India
December 2018

Abstract

This report describes the approaches adopted to solve the problem whose description goes as follows:

"Concept extraction from legal document text and relationship identification between identified concepts using the predefined Ontology".

We have legal documents from the client as well as the Ontology defined by the expert in the domain. Our goal is to build an algorithm to identify the concepts from the sections of the document and with the help of predefined Ontology, we have to extract the relationships between the identified concepts.

This report describes the approach taken to solve the first sub-part of this problem, i.e. Concept Extraction from the Legal Document. To solve this, we have proposed two Probabilistic and two Rule-Based approaches in this report.

Contents

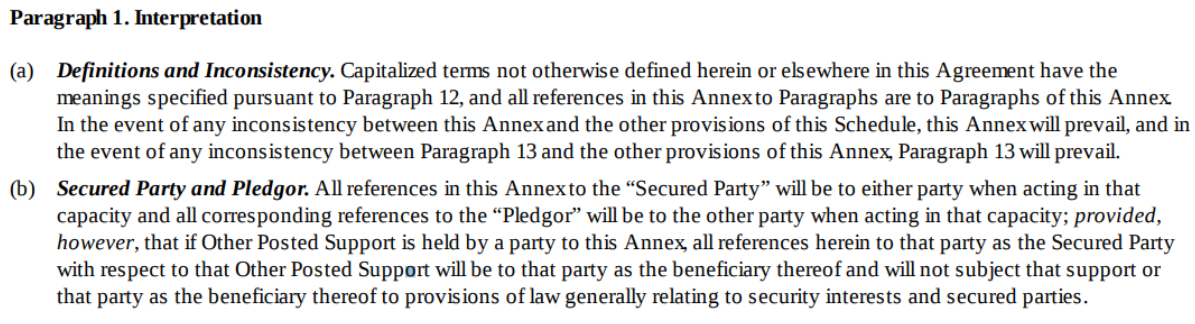
Chapter 1

Introduction

1.1 Overview

Legal Documents are known to be concise in representing information, complex in structure and difficult to understand. The size of one document may vary from few pages to hundreds of them. They express conditions in Natural Language form that describes a certain set of actions or rules in the context they regulate and understanding them manually might be very time consuming and tiring task to do for a normal Human Being or a Lawyer.

Following figure shows a paragraph taken from a legal document to understand the level of complexity it is comprised of.



Paragraph 1. Interpretation

(a) ***Definitions and Inconsistency.*** Capitalized terms not otherwise defined herein or elsewhere in this Agreement have the meanings specified pursuant to Paragraph 12, and all references in this Annex to Paragraphs are to Paragraphs of this Annex. In the event of any inconsistency between this Annex and the other provisions of this Schedule, this Annex will prevail, and in the event of any inconsistency between Paragraph 13 and the other provisions of this Annex, Paragraph 13 will prevail.

(b) ***Secured Party and Pledgor.*** All references in this Annex to the "Secured Party" will be to either party when acting in that capacity and all corresponding references to the "Pledgor" will be to the other party when acting in that capacity; *provided, however*, that if Other Posted Support is held by a party to this Annex, all references herein to that party as the Secured Party with respect to that Other Posted Support will be to that party as the beneficiary thereof and will not subject that support or that party as the beneficiary thereof to provisions of law generally relating to security interests and secured parties.

Figure 1: Text from Legal Document

Realising the requirement for an Automatic facility supporting this intellectual activity which could help us extract key concepts from the documents and estabhlish relationship between them in a stint of time, we put forward a few approaches to reduce this manual work and propose Natural Language Processing (NLP) based methods along with the results we obtained after employing them.

1.2 Organisation of the Report

As described in the abstract, the problem statement involves two parts and this report presents the approaches adopted to solve the first part of this problem. These approaches have been described in Chapter 2 of this report. Later in Chapter 3, we conclude this report by analyzing why some pf the proposed approaches didn't work for us while the other approaches did. Along with that, we also propose some other approaches as future work that could be experimented upon to tackle sub-problem 1.

Chapter 2

Implementation

2.1 Sub-Task 1: Text Extraction from PDF

The text data present in the PDF document file was converted to JSON[1] Format using PDFMiner and Document Reader Tool. One thing that should be noted here is that each section's head might either be composed of a paragraph or bullets points. Thus the extraction part was done by ensuring that all the text falling under a particular section head is extracted as a single unit.

Once the JSON File is prepared and the whole text has been extracted from the PDF, we move on to apply a further level of text extraction and store the text as class objects which is finally stored as a pickle so that we can reduce the overhead of iterating over the JSON again and again to extract the text.

2.2 Sub-Task 2: Concept Extraction

This section briefly describes the approaches that were adopted to extract concepts from the documents available to us. These include both Probabilistic as well as Rule-Based approaches.
We have two approaches for both probabilistic and rule-based methods.

2.2.1 Probabilistic Approaches

Step1: Data Pre-Processing

As concluded in Section 2.1, we have stored the text data extracted from the JSON as a class object, one of the variable of which stores each paragraph of the document as an element of a list.
We apply the following pre-processing techniques before moving on to the concept extraction part:

1. We concatenate the whole list of paragraphs into a single string.
2. The whole text is converted to lowercase.
3. We remove the stopwords, numbers and other irrelevant characters from the text.

Step2: Concept Extraction

- **Approach 1:**

  Once we are done we with the pre-processing part as mentioned in Step 1, we move on to form chunks or n-grams from the text thus obtained. The intuition behind doing this is that each concept might appear as a single word or a group of words that should appear adjacently in the text. Thus, we form all such unigrams, bigrams, trigrams and fourgrams using NLTK's[2] predefined functions.

An n-gram is defined as a contiguous sequence of n items from a given sample of text.

Now, as our discussion is revolving around probability, so we shall find how often a particular ngram appeared in the whole document. This can be easily done using either the Python's Counter class or NLTK's FreqDist method. In order to convert these counts to probabilities, we find the number of n-grams of each type(uni/bi/tri/four) and divide each n-gram's count with the total count of ngrams of its type.

E.g. Consider a trigram "X Y Z". Let us assume it appeared n number of times in the text and the total number of trigrams be N. Then, we have

$$p(\text{"}XYZ\text{"}) = (n/N)$$

The probabilities thus obtained are sorted in decreasing order and higher is the more probability, higher would the appearance of that n-gram in the document.

Results:

Ngram	Score
('secured', 'party')	117.0
('credit', 'support')	87.0
('years', 'years')	84.0
('years', 'greater', 'years', 'years')	81.02

Shortcomings of Approach 1:

This approach does not consider any information about a concept and it simply assigns a score to each word or pair of words based on its occurrence. Thus, there are high chances of irrelevant pairs being extracted and ranked high just because they appeared a lot of times in the document.

- **Approach 2:**

  This is an extension of the approach mentioned above. Here, we shall not only be looking at the occurrences of a n-gram but we shall also find how the extracted pairs of words were associated to each other. This information is known as PMI[3] or Point-wise Mutual Information.

  The purpose behind using PMI as a measure to find the concepts in our document is that it looks for the association between words. So, it is highly probable that a particular concept appeared only once in the whole document and its constituents(words) didn't appear anywhere else in the document. So if we go by approach 1, this concept will be assigned a very low probability though it was a proper concept in context of document. But if we use PMI as a measure to rank the concepts, such low-frequency concepts can still be captured easily.

  Similar to what we did in Approach 1, we still find all the unigram, bigram, trigram and four-gram probabilities and add an extra step after that to calculate the PMI score for an n-gram. E.g. Consider a bigram "X Y". Its PMI is calculated as follows:

$$PMI("XY") = P("XY")/(P("X") * P("Y"))$$

  where function P is same as that defined in Approach 1.

  The PMI scores thus obtained are sorted in decreasing order. Higher is the score, higher would the possibility of an ngram being a concept.

  Results:

Ngram	**Score**
('consult', 'attempt')	8.58
('taking', 'arithmetic')	8.57
('assure', 'safe', 'custody')	17.17
('uniform', 'commercial', 'code')	17.16

Shortcomings of Probabilistic Approaches:

Though our probabilistic approaches were capable of extracting the concepts from the document to some extent and there still appeared a lot of irrelevant concepts in the final extracted list. This happened due to the high occurence of its constituent tokens in the text and here, we conclude that probabilistic approaches are not expected to help us solve the task of concept extraction.

2.2.2 Rule-Based Approaches

Having implemented and observed two probabilistic approaches, we reached to this conclusion that these are not suitable for solving our task.

After closely observing the document, we see that the concepts majorly appear as Noun Phrases in the document and this introduces a scope for trying Rule-based approaches wherein we shall utilize certain grammar rules keeping in mind the structure a legal document pertains to.

In this section, we'll be formulating certain patterns of POS tags to extract the concepts.

Step1: Data Pre-Processing

Similar to what we began our pre-processing with in the probabilistic approaches, we begin with that same pickle file here also but we don't apply any pre-processing techniques on this. The proposed rule-based techniques will be built on POS tags and thus we can't remove stop words, numbers etc. from our text. Therefore, we'll be preserving the original structure of our document which we had extracted and stored as a pickle.

NOTE: We use Spacy[4] for POS Tagging.

Step2: Concept Extraction

- **Approach 1:**

  Reading a particular paragraph in the document and analyzing the patterns of POS tags appearing around the concepts led us to formulate 4 grammar rules. These are:

Pattern	Example
DET-PROPN	The Plegdor
DET-VERB-PROPN-PROPN	all Posted Credit Support
DET-PROPN-PART-PROPN-PROPN	The Pledgors Transfer Amount
DET-PROPN-PROPN	The Delivery Amount

  NOTE: In the above table, DET, PROPN and VERB denotes Determiner, Proper Noun and Verb respectively.

  Once we have extracted chunks out of text on the basis of the above mentioned rules, we move to extract the noun phrases from them. Each pair of co-occuring Proper Nouns are treated as a Noun Phrase, i.e. in the above examples, "*Delivery Amount*" is treated as a Noun Phrase and therefore as a concept also.

- **Approach 2:**

  One shortcoming of Approach 1 mentioned above is that it involves manual observation of grammar rules in the parahraphs. Thus, it is possible for us to miss out some concepts in the document just because we missed the grammar for it. To avoid such cases, we extend the above approach and use Spacy's Noun Chunk extractor. When tested on one paragraph, it is found that the noun chunk extractor not only extracts the noun phrases that were extracted from Approach 1 by manually deciding the rules, but it is also able to cover some extra cases which we might have missed out in manual formulating the rules.

Chapter 3

Conclusion and Future Work

In this report, we closely observed and analyzed some probabilistic and rule-based approaches to extract concepts from legal documents.

Probabilistic approaches didn't prove out to be much helpful to extract concepts from these type of documents.
This occurred due to the following reasons:

1. Probabilistic approaches take into account the occurrence of items to generate their results but it is not at all necessary for a concept to appear many times in the document. By this, we mean that for a paricular set of words to be flagged as a concept will not at all depend on its frequency.

2. Legal Documents obey certain structure. We simply extracted all the text from the document and didn't utilise the structural properties.

The reason why Rule-Based approach proved out to be an apt approach for our task is because of they were modeled to capture the structure of sentences in which a concept appears and they were also independent of probabilities of these phrases.

3.1 Future Work

Though the rule-based approach worked perfectly fine in this scenario and can be employed to any such similar task, still, as a future work, trying deep learning based approaches can be an area to work upon. Due to the inavailablity of large amount of legal documents, we were unable to experiment in this direction but if sufficient data is available, a deep learning based approach might work even better than rule-based approach.

This report describes our approach to only subproblem-1 which involves concept extraction from the documents. Once we have extracted the concepts, we are also supposed to find relationship between these concepts by using the predefined Ontology.

Bibliography

[1] "Json (javascript object notation)." [Online]. Available: https://www.json.org/

[2] E. L. Bird, Steven and E. Klein, "Natural language processing with python." 2009.

[3] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction. in proceedings of the biennial gscl conference, potsdam, germany." 2009, pp. 31–40.

[4] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," 2017.