SAIDL INDUCTION ASSIGNMENT
PART 1

PROBLEM STATEMENT:

Automation of extraction of rules from regulatory texts.

DESCRIPTION: This problem mainly pertains to firms from the financial and business sector, but can be extended to other fields as well. An important function of a certain department of these firms is to construct precise business rules from large pieces of regulatory texts related to compliance or obligation issues. An example is stated here:

EXCERPT FROM A REGULATORY ARTICLE:

Report in column A the net investments in all leases to individuals for household, family, and other personal expenditures (i.e., consumer leases). Include direct financing leases accounted for under ASC Topic 840, Leases, by an institution that has not adopted ASC Topic 842, Leases; direct financing and sales-type leases accounted for under ASC Topic 842 by an institution that has adopted this topic; and leveraged leases accounted for under ASC Topic 840 (including those that were grandfathered upon the adoption of ASC Topic 842 and remain grandfathered). For further information on extending credit to individuals for consumer purposes, refer to the instructions for Schedule HC-C, part I, items 6.c, "Automobile loans," and 6.d, "Other consumer loans."

BUSINESS RULE EXTRACTED BY A DOMAIN EXPERT FROM THIS ARTICLE:

Report in column A the net investments in all leases to individuals for household, family, and other personal purposes. They include direct financing and sales-type leases accounted for under ASC Topic 842 by an institution that has not adopted this topic.

The task of constructing such business rules from the regulatory documents is very labour-intensive if done manually. It is also more prone to human errors arising out of haste, tiredness and loss of concentration. The problem statement here is to devise a method or model to somehow automate this process of constructing business rules from large chunks of regulatory documents.

LITERATURE REVIEW:

Similar work is also being done in the legal domain where the objective is to extract rules from legal documents. Specifically the objective in the use case studied was to identify and extract deontic components (clauses exhibiting permission, obligation or prohibitions) and construct rules based around these three contexts. A paper that was studied in depth for understanding the process implemented for carrying out this objective is:

Dragoni et.al. Combining NLP Approaches for Rule Extraction from Legal Documents

A brief concept-note of the procedure being followed to attain the objective in this paper is presented below:

CONCEPT NOTE

- ‘Deontic’ means something related to duty and obligation. It is difficult to apply deontic reasoning in real world ethical scenarios.
- This is because legal documents are not written in machine processable language, we have to work with natural language texts.
- Our aim is to automate the extraction of a set of rules from a legal document written in natural language.
- 2 main problems that need to be tackled in this regard are:
    - To deal with the variability of natural language texts for identification of deontic components of each rule.
    - To combine a **syntax-based** approach with a **semantic-based** one to identify the terms that constitute each rule and correctly assign them as an antecedent/consequent of the rule.
- Our primary tools for this approach are:
    - A deontic lightweight ontology describing the deontic linguistic elements and allowing the identification of **obligations, permissions, prohibitions**. ***(O1)***
    - A text-structure lightweight ontology describing the structure of the natural language and how punctuations can be interpreted to help the rule extraction process. ***(O2)***
    - *Stanford Parser:* for a syntax-based rule extraction procedure. ***(SP)***
    - *CCG Parser tool:* for a logic-oriented rule extraction procedure. ***(CCG)***

THE FRAMEWORK

NATURAL_LANGUAGE_FILE.pdf

--- PDF is parsed

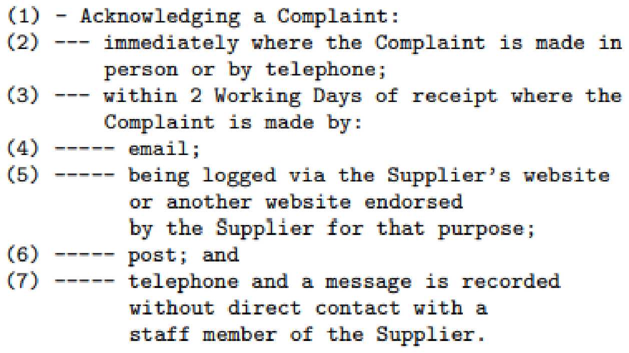
NATURAL_LANGUAGE_FILE.txt

--- **O1:** Automated identification of sentences expressing obligation, permission, prohibition by segregating into respective subclasses.

--- **O2:** Case specific ontology to analyze text structure and punctuation. In our case, 3 concepts have been modelled for analysis: (i) Document: pertaining to the entire text, (ii) TextChunk: pertaining to a single block of text and (iii) Punctuation

EXTRACTION OF SENTENCES OF INTEREST

Exercise O1 and O2 on single text chunks which results in the formation of a linked tree like structure as shown:



```
(1) - Acknowledging a Complaint:
(2) --- immediately where the Complaint is made in
        person or by telephone;
(3) --- within 2 Working Days of receipt where the
        Complaint is made by:
(4) ----- email;
(5) ----- being logged via the Supplier's website
          or another website endorsed
          by the Supplier for that purpose;
(6) ----- post; and
(7) ----- telephone and a message is recorded
          without direct contact with a
          staff member of the Supplier.
```

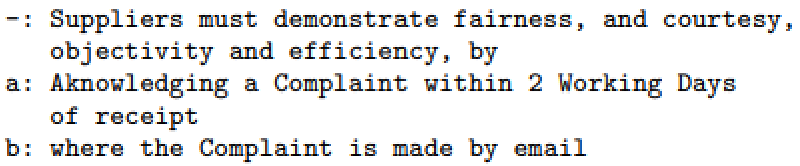
Such type of mapping assigns Level 1 to (1); Level 2 to (2) and (3) and Level 3 to (4)-(7). Various sentences extracted from the text are the conjunctions of the following chunks: (1)-(2); (1)-(3)-(4); (1)-(3)-(5); (1)-(3)-(6); (1)-(3)-(7). Punctuations are used as regulators to split complex sentences. Single terms are later extracted from these sentences.

SENTENCES EXTRACTED FROM TEXT

Application of ***SP***: includes a Tree Parser that produces a tree-based representation of each sentence. It works out grammatical structure of sentences based on the English language syntax. Uses knowledge gained from manually parsed sentences to analyze new ones. Probabilistic in nature, but works well for most of the cases.

EXTRACTION OF TERMS

A ***term*** is a complex text expression representing an entire concept. In general, the beginning of a new subordinate sentence is considered a new term. For example:

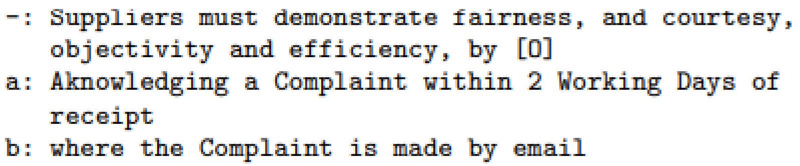


```
-: Suppliers must demonstrate fairness, and courtesy,
   objectivity and efficiency, by
a: Aknowledging a Complaint within 2 Working Days
   of receipt
b: where the Complaint is made by email
```

The first row is considered ‘implicit’. The two identified terms by the parser are a. and b.

ANNOTATION OF TERMS WITH DEONTIC TAGS

Extraction of terms is followed by their annotation with deontic tags : suppose (O) for Obligation; (Pe) for Permission and (Pr) for Prohibition. If one of the lemmatized version of the labels of the vocabulary is present in the sentence, the term is annotated with the respective tag. For Example:

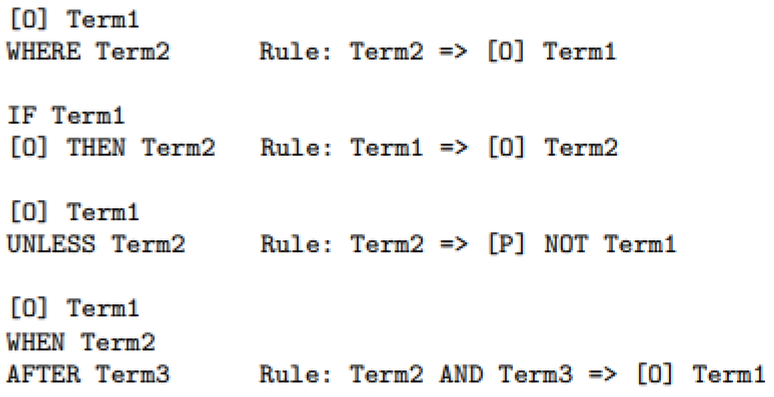


```
-: Suppliers must demonstrate fairness, and courtesy,
   objectivity and efficiency, by [O]
a: Aknowledging a Complaint within 2 Working Days of
   receipt
b: where the Complaint is made by email
```

Obligatory due to the presence of “*must*”.

COMBINATION OF TERMS FOR RULE DEFINITION

Rules are defined by combining the extracted and the annotated terms. For creating the rules, a set of patterns is applied to detect the antecedent and consequent of each rule. Some of these patterns are:



```
[O] Term1
WHERE Term2       Rule: Term2 => [O] Term1

IF Term1
[O] THEN Term2    Rule: Term1 => [O] Term2

[O] Term1
UNLESS Term2      Rule: Term2 => [P] NOT Term1

[O] Term1
WHEN Term2
AFTER Term3       Rule: Term2 AND Term3 => [O] Term1
```

In case a deontic tag is used for annotating an implicit term, such a tag is inherited by the first term following the implicit one. Hence in our case we will apply the first pattern due to presence of the “where” label and the generated rule will become: **b => [O] a.**

USAGE OF THE CCG PARSER

*The **CCG***: Integrated to perform a logical analysis of each sentence in order to establish relationships between words. Its main aim is to support the NLP pipeline constructed using SP alone, for instance in cases where the SP is not able to extract much info using its pattern mechanism. Improves the general effectiveness of the rule extraction system.

FINAL SET OF RULES EXTRACTED

EVALUATION OF THE APPROACH

- Evaluation was done on the Australian Telecommunications Consumer Protections Code.
- The methodology used was to compare the auto generated set of rules with the set of rules manually generated by an analyst.
- The broad parameters used for evaluation were:

    i) Number of correct sentences extracted from the text,

    ii) Number of correct terms identified in the extracted sentences,

    iii) Number of correct deontic components annotations performed on the identified terms,

    iv) Number of correct rules generated from the extracted terms annotated with the deontic component, and

    iv) Impact of the CCG parser in supporting the detection of new patterns for generating rules:

    - Agreement between the rules extracted from the CCG output and the ones generated by the SP pipeline, and
    - Number of rules correctly extracted from the CCG output regarding sentences for which the lower branch generated anything.

- Validation scores were thus calculated. Errors arise mainly due to incorrect identification of propositions in the first step and the relative positioning of the subject and predicate of sentences.

- Extraction of all synsets related to Obligation, Permission and Prohibition enriches the efficiency of the ontologies used.

The current task at hand is very similar to the process of abstractive text summarization. In abstractive text summarization, a model is trained on large text chunks and is tasked with summarizing the corpus into a smaller version of itself ; including only the most important highlights. It is different from extractive text summarization in the way that in extractive summarization, the algorithm extracts the important sentences directly from the text and arranges them accordingly, hence there are chances that the resulting summary might be grammatically or semantically incorrect. However in the abstractive process, the algorithm is built in such a way that it creates the summary in its own words (paraphrases) and hence reduces the chances of the occurrence of grammatical and semantic inconsistencies.

INGREDIENTS:

>> Training dataset of text corpus written in natural language along with the rules extracted from them by a domain expert.

>> Test dataset of text corpus written in natural language to be fed to the model.

Rules extracted from these texts by domain experts are also required for validation of results.

>> A summarization model that could perform abstractive text summarization on text written in natural language

>> A validation model to quantify the correctness of the output obtained from the summarization model.

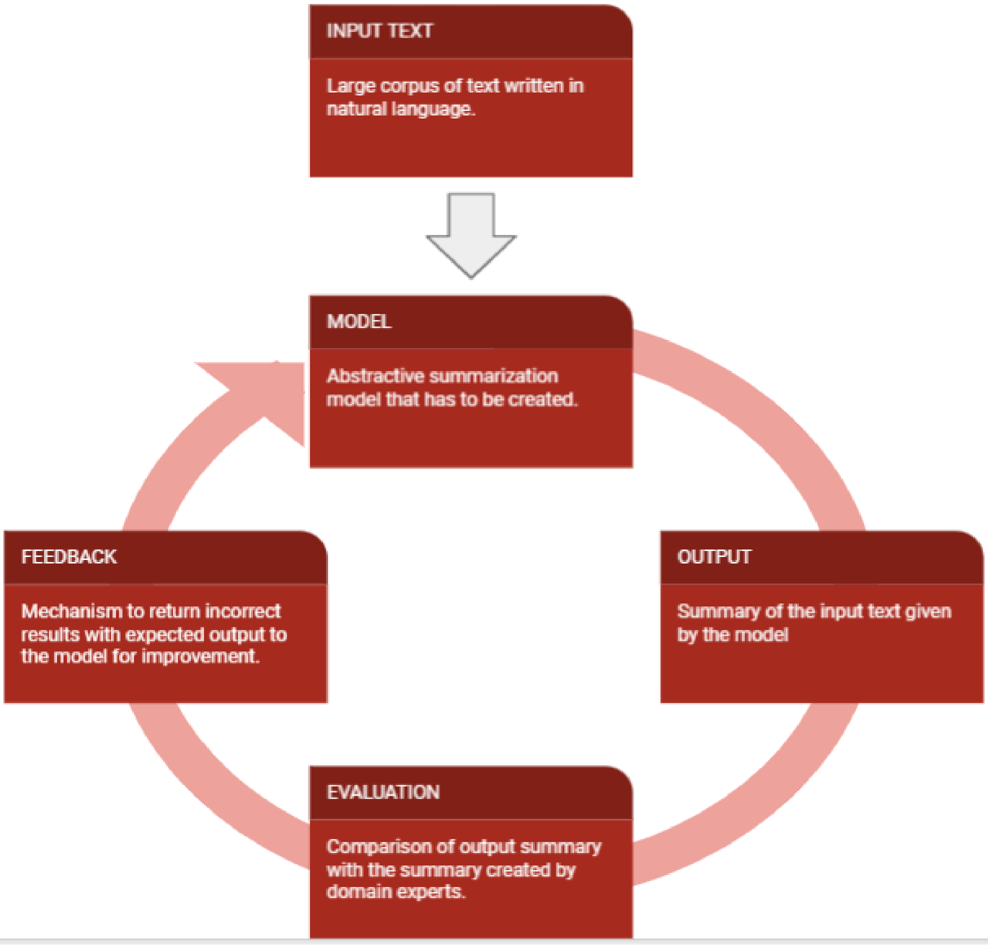
TOOLKIT:

>> Python libraries for natural language processing: nltk, spacy

>> HuggingFace offers a series of pre-trained models for summarization and various other purposes. The architecture of these models could be used as a reference.

>> Amazon sagemaker or google cloud services can be used for carrying out the model training.

MODEL DRAFT:



INPUT TEXT

Large corpus of text written in natural language.

MODEL

Abstractive summarization model that has to be created.

OUTPUT

Summary of the input text given by the model

EVALUATION

Comparison of output summary with the summary created by domain experts.

FEEDBACK

Mechanism to return incorrect results with expected output to the model for improvement.

MODEL EVALUATION:

>> An example proposal for validation could be: Store the output given by the summarizer as a tokenized list. Create another tokenized list from the rule extracted by a domain expert. Find the cosine similarity between these two lists and quantify the correctness of output by finding the cosine similarity score. Define a threshold (80%-85%) for the score below which the output should be deemed unsatisfactory.

>> Design a feedback mechanism to feed the unsatisfactory outputs back to the model along with expected results and try to implement a way to make it learn better.