TATA CONSULTANCY SERVICES
Experience certainty.

TATA

Implementation-centric Classification of Business Rules from Documents

Preethu Rose Anish, Abhishek Sainani, Abul Ahmed, Smita Ghaisas
TCS Research
Pune, India

Introduction

- In large multi-site multi-vendor projects
  - studying requirement documents to understand problem domain and inferring possible solution
    - an important activity in Requirements Engineering
  - The process of reading User requirements Specification (URS) to create Software Requirement Specification (SRS) is knowledge intensive activity
  - Automated Interpretation of the URS in terms of implementation-specific knowledge elements for software engineers' consumption has been reported in the past
    - aim of such an interpretation is to reduce the effort associated with a manual extraction of knowledge elements

Our Contribution

- A deep learning model for an implementation-centric classification of one such knowledge element, namely, business rules.
    - An approach based on a Bidirectional Long Short Term Memory Network (BiLSTM) to capture the context information for each word
    - followed by an attention model to aggregate useful information from these words to get to the final classification.

Our model adopts an end-to-end architecture that does not rely on any handcrafted features

Business Rules

- From an information-system perspective,
  - a Business Rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business.
- guide operations of businesses
- An important source of information for software system implementation
- often found to exist in the form of both structured and unstructured text in requirements documents and other project documentation such as business agreements, charter documents, feature descriptions, guidebooks and technical specifications.
- Often found embedded in descriptions of business processes, use cases and workflows; rather than explicated independently.
- Even when documents contain dedicated sections on business rules, they are likely to be found scattered across other locations making their comprehension a tedious task.

Classification Schema for Business Rules

- Rule acts to present an implementation-centric classification of business rules.

- **Rule acts:** composed of frequently co-occurring Rule intents.

- **Rule intents:** atomic constraints embedded in a natural language rule statement.
  - Each rule statement can therefore be represented as a composition of Rule intents.

- We then applied a clustering algorithm over the rule statements to detect the Rule intents that co-occur most frequently.
  - The groups of frequently co-occurring intents were manually inspected to name the various "Rule acts"

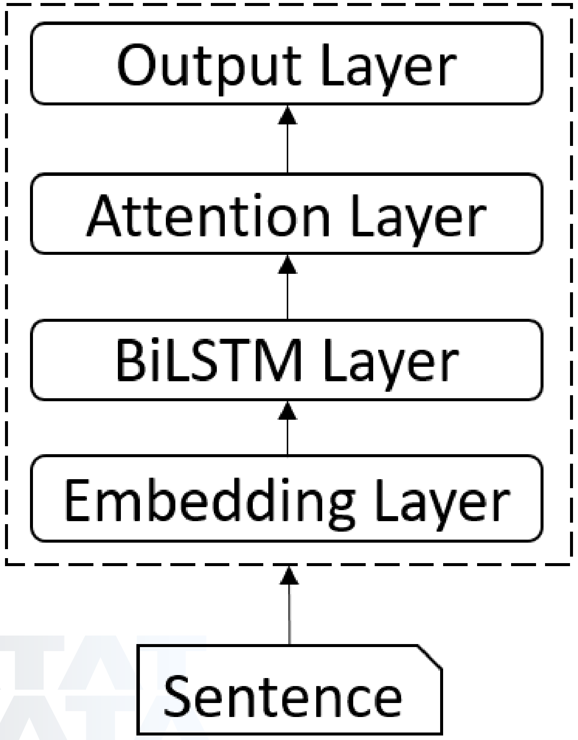
Our Approach - Dataset

- Total 4043 sentences
    - 1676 business rule and 2367 non-rules.
- Taken from 40 SRS documents from large projects in the Insurance domain.
    - Annotations already available for 30 documents (from our previous work).
    - We annotated 10 more SRS documents (owing to Deep learning data demands).
        - Done by one in-house subject matter expert

Classification Architecture

- Two-step classification method
  - First classifier identifies business rules
  - Second classifier assigns one or more Rule acts to the identified business rule statement

- Classifier architecture – BLSTM-att,



Classification Architecture

- **Embedding Layer**
  - Word2Vec model (Skipgram with negative sampling)
- **BiLSTM Layer**
  - Use BiLSTM instead of LSTM to learn contextual dependencies among words. This provides BiLSTM with the complete context for a given word
- **Attention Layer**
  - Instead of relying on information from the last node, to obtain information at every node of the mode
- **Output Layer**
  - Sigmoid function
- **Loss Function**
  - binary cross-entropy loss or Sigmoid cross-entropy loss

Experiments and Results

- Used a 5-fold cross validation technique to test performance
- First Classification Task – Experiment 1
  - Training data – set of both rule and non-rule statements
- Second Classification Task – Experiment 2
  - Training data – set of rule sentences

Experiment 1 Results

	Precision (in %)	Recall (in %)
Baseline	84	78
BLSTM-att	88.11	87.6

Experiment 2 Results

Class (Rule acts)	Precision (in %)	Recall (in %)
Deadline	91.65	74.99
Conditional Execution	84.09	77.89
Access Control	74.28	83.87
User Interface	92.47	80.95
Data Validation	55.36	49.02
Data Protection	99.9	49.9
Documentation Mandate	72.72	44.44
Calculational	71.42	80
Financial transaction	62.96	56.66
User Responsibility	70.69	74.54

Comparison with Baseline

Class (Rule acts)	Baseline Precision (in %)	BLSTM-att Precision (in %)
Deadline	72.11	91.65
Conditional Execution	60.94	84.09
Access Control	52.54	74.28
User Interface	91.8	92.47
Data Validation	92.65	55.36

Discussion

- Experiment 1 (binary classification) and Experiment 2 (multilabel classification) clearly indicates that a BLSTM-att model performs better as compared to our previous work that used linguistic pattern based method

- *Data Validation* - includes a broad spectrum of information that can itself be categorized into sub-categories

- *Documentation Mandate* and *Data Protection* have low recall, mainly due to low number of instances in the business documents that we used in our experiments

Conclusion

- Requirements Engineering phase of software development involves an effort intensive activity of reading URS to create SRS.

- Interpreting URS in terms of implementation-specific knowledge elements aids in
  - reducing the effort associated with a manual extraction of knowledge elements and subsequently,
  - their "translation" into primitives understood by those who must build the intended software.

- Presented a deep learning architecture for extracting business rules from URS and classifying them into different implementation-centric classes.

- Obtained a precision and recall of 88.1 % and 87.6% respectively for the binary classification task
  - a decent improvement over our baseline approach where we obtained a precision of 84% and a recall of 78%

- Similarly, for the multi-label classification as well we see an improvement in scores except for *Data Validation*

Future Work

- consider more advanced LSTM variants:
  - deeper self-attention mechanisms
  - stacking BILSTMs [26]
  - pre-training the BILSTMs with auxiliary tasks

- Explore pre-trained language models
  - ULMFiT
  - BERT

Thank You !