Title: Understanding the Logical and Semantic Structure of Large Documents

ABSTRACT

Up-to-the-minute language understanding approaches are mostly focused on small documents such as newswire articles, blog posts, product reviews and discussion forum entries. Understanding and extracting information from large documents such as legal documents, reports, proposals, technical manuals and research articles is still a challenging task. The reason behind this challenge is that the documents may be multi-themed, complex and cover diverse topics. For example, business opportunities may contain information on the background of the business, product or service of the business, plan, team management, financial or budget related data, competitors, logistics, compliance, legal information and boilerplate content that is repeated across documents. The content can be split into multiple files or aggregated into one large file. As a result, the content in the whole document may have different structures and formats. Furthermore, the information is expressed in different forms such as paragraphs of text, headers, data forms, tables, images, mathematical equations, lists or a nested combination of these structures.

Neither all documents follow a standard sequence of sections nor they have a standard table of contents. Even if a table of contents is present, it is not straightforward to map a table of contents across documents and a table of contents doesn't map documents section headers and subsection headers directly. Moreover, not all documents from the same vertical domain have consistent section headers and subsection headers.

Semantic organization of sections and subsections of documents across all vertical domains are not the same. For example, a business document has a completely different structure from a user manual. Even research articles from computer science and social science

have completely different structures. For example, social science articles have methodology sections where as computer science articles have approach sections. Semantically these two sections should be same.

Identifying a document's logical sections and organizing them into a standard structure to understand the semantic structure of a document will not only help many information extraction applications but also enable users to quickly navigate to sections of interest. Such an understanding of a document's structure will significantly benefit and inform a variety of applications such as information extraction and retrieval, document categorization and clustering, document summarization, fact and relation extraction, text analysis and question answering. It will help to extract different important elements such as tables, headers, footnotes, biographies and images, which can be directly useful to other text analysis tasks. Human are often interested in reading specific sections of a large document and hence will find semantically labeled sections very useful. It will help human being to simplify their reading operations as much as possible and save their valuable time.

We intend to develop a framework that can analyze a large document and can provide a view, which helps human being to know where particular information is in that document. For example, we have a report on university network security and we want to know what policies have been employed by the security team to prevent attacks. The framework will point us to the sections that describe the policies in the given report. Another example, we have a request for proposal (RFP) of 100 pages and we want to know the boilerplate available in that RFP. The framework will point us to the appropriate sections that represent the boilerplate in a given RFP. In a nutshell, we aim to automatically identify and classify semantic sections of documents and assign human-understandable and consistent labels to similar sections across documents.

We are using RFP documents for our experiments. RFPs are usually very complex documents and have lot of noisy contents. We have cleaned, preprocessed, normalized and

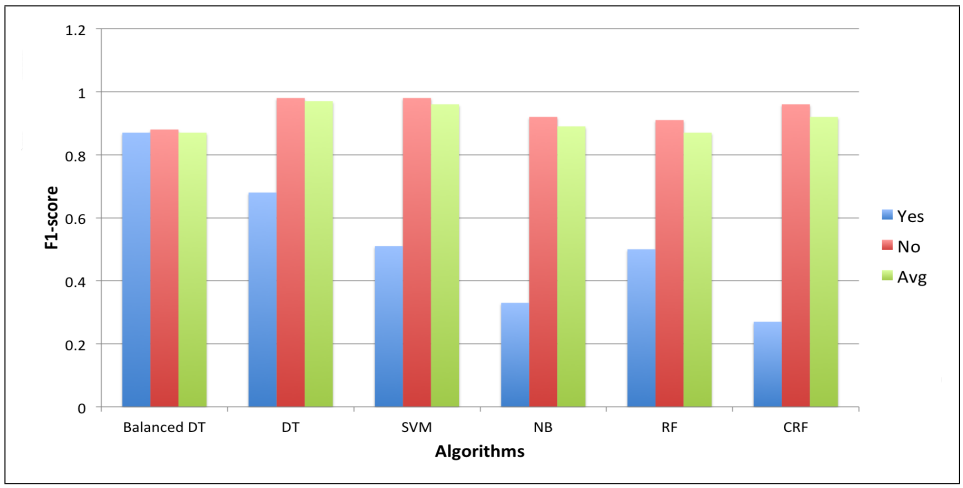

FIG. 1. Comparison of f1-score for different algorithms [Balanced DT = Balanced Decision Tree; DT=Decision Tree; SVM = Support Vector Machines; NB=Naive Bayes; RF = Random Forests; CRF= Conditional Random Field]

identified sentence and paragraph level boundaries. We have also generated named entities and Part-Of-Speech tagging for sentence level. Then we have stored all information in Elasticsearch.

We are developing powerful, yet simple, approaches to build our framework using layout information and text content. Layout information and text are extracted from PDF documents such as RFP documents. Our framework has four units: Pre-processing Unit, Annotation Unit, Classification Unit and Semantic Annotation Unit.

In the classification unit, we have developed classifiers for section header identification. We used layout and contextual information for feature generation. We developed our classifiers using Support Vector Machine(SVM), Decision Tree, Conditional Random Field(CRF), Naive Bayes, Long Short-Term Memory(LSTM) Neural Network and Random Forests algorithms. We analyzed the performance of all models developed by these algorithms and chose the best model. Figure 1 compares results for Decision Tree, SVM,

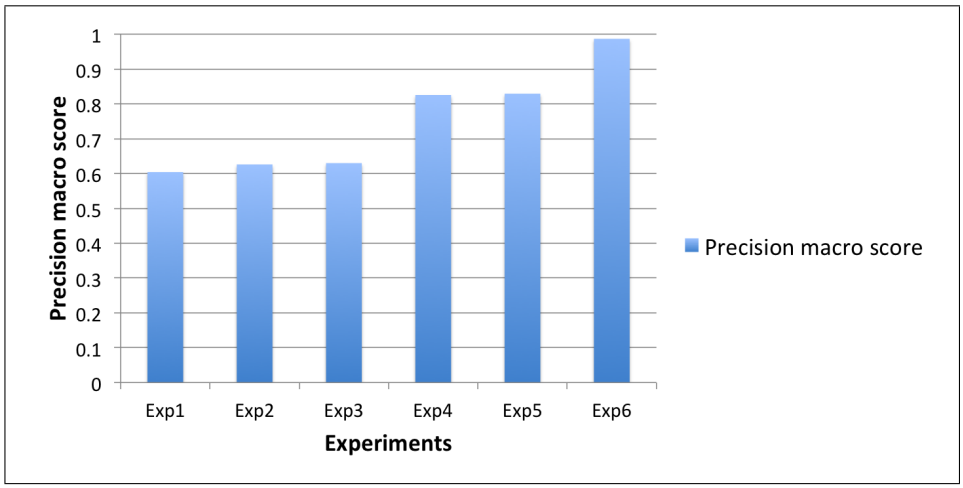

FIG. 2. Comparison Over Features and Experiments

CRF, Naive Bayes and Random Forests algorithms. As many RFPs are not well structured, we haven't achieved very good results for section header classifiers. We are working more on section header classifiers to improve accuracy and to make classifiers more generic for different kinds of large documents. The output of section header classifiers will be used as input for section classifiers. Section classifiers will identify different levels of section header such as top level section, subsection and sub-subsection header. The section classifiers will have a section boundary detector which will split document into physically divided sections. The divided sections will be input to the semantic annotation unit for human understandable semantic labeling.

We have also developed classifiers to identify boilerplate from RFPs. The boilerplate detection classifiers are used for removing unnecessary content from RFPs. We started with identifying sentence-level boilerplate and ended with paragraph-level boilerplate. We have done the necessary number of experiments with different features and configurations to get an acceptable result for boilerplate detection. We have achieved 98.68% accuracy

for paragraph-level boilerplate detection on training data and 94.43% accuracy on test data. Figure 2 shows precision macro scores for different experiments on training data. We used scikit-learn machine learning framework for tokenization, feature extraction, model generation and evaluation.

We will explore and experiment with more machine learning approaches such as deep neural networks for semantic section identification within a document and labeling them with semantic names. We will use techniques such as LSTM and assess their contributions using feature ablation studies.

We will develop a generic and domain independent framework. Though for the experimental purposes, we are using RFP documents, extended experiments will be done using medical documents and academic scholarly articles.