The Serialization of Heterogeneous Documents

Peter John Hampton, William Blackburn, Hui Wang
Artificial Intelligence and Applications Research Group
Ulster University, Jordanstown
United Kingdom, BT37 0QB
Email: hampton-p1@email.ulster.ac.uk
{wt.blackburn, h.wang}@ulster.ac.uk

Abstract—Tasks involving the analysis of natural language are typically conducted on a corpus or corpora of plain text. However, it is rare that a document is unstructured and freeform in its entirety. Documents such as corporate disclosures, medical journals and other knowledge rich archive contain structured and loosely-structured information that can be used in a variety of important text mining tasks. In this paper we propose a syntactical preprocessing architecture to serialize presentation-oriented documents to a machine readable format that aspires to preserve the document structure, contents and metadata. We introduce a hybrid pipeline architecture, discussing the various processes and the future research direction that could potentially lead to a holistic representation of heterogeneous documents.

I. INTRODUCTION

Knowledge mining researchers and practitioners have been implementing techniques to aid and enact decision-making from knowledge discovery tasks. However, various challenges restrict the computational understanding of language found in such documents as analysis has traditionally focused on plain text (1; 2; 3). This paper proposes a hybrid preprocessing architecture for preserving a documents contents in its entirety and converting selected entity classes to their canonical form, enabling deeper analysis.

Although there are arguably many documents of interest, we focus the attention of this paper on corporate disclosures, specifically interim financial reports (10-Qs) due to the depth of knowledge and the complexity of their composition. We demonstrate, at a high level, a multistage architecture in Fig. 1 that combines both statistical and rule based approaches for serialization to preserve the documents structure and content.

Diverse developments in Information Retrieval methodology have been made over the past two decades, which could make it sufficiently easier to represent unpredictable document formats and associated contents. We describe related work and motivations behind this research in Section II. Section III analyzes the document structures of five company interim disclosures. In Section IV we compare various mainstream data serialization formats while concluding the advantages and disadvantages among them. Section V describes the pipeline components depicted in Fig. 1 in substantive detail, breaking down each process into a set of processes. The paper concludes in Section VI which sets a future direction for our research.

Fig. 1. The Hybrid API Architecture described in this paper for the syntactic representation of heterogeneous documents. At a high level, the depicted analysis pipeline is broken down into 6 stages: Detection (a), Acquisition (b), Segmentation (c), Entity Morphing (d), A cache object for document representation (e) and a duel error handling module (f). A document is introduced into the pipeline and subject to numerous decomposition stages.
II. RELATED WORK

Early attempts at analyzing structured data in heterogeneous documents include the Douglas et al study (4) which represented tabular data using spaces in plain text. These researchers presented an analysis of table layout and the associated linguistic characteristics. However, wider linguistic complexities are presented in plain text representation of tables. This can include complex header parsing, redundant or null cell representation and wider linguistic complexities. We aim to extend their problems of characteristic syntax and apply a set of transformations to the contents.

More recent work includes the Clark & Divvala research (5) which aims to discover a holistic view of the document, achieving a deeper semantic understanding of articles such as academic computer science papers by placing emphasis on figures such as charts. Likewise, the data found in corporate disclosures tend to be multifaceted in nature, that being a mix of unstructured, loosely-structured, unstructured text and miscellaneous data. Their open-source solution agnostically parses a document and locates the areas wherein figures or tables could reside by reasoning about the empty regions within that text, achieving success due to its relaxed formatting assumption.

III. BACKGROUND & ANALYSIS

In this paper we refer to freeform text, that is sentences with no predictable syntactical structure as unstructured content. In turn, we refer to information in tabular form as structured data. Although the term ‘loosely-structured data’ is typically used to refer self-describing data (6; 7), we use it in this paper as a means of classifying information that is not presented in a structured or unstructured form. This can take the form of bullet points, footnotes, images etc.

We analyzed a random sample of five interim statements published by different software companies between July and December 2014 as listed on the British Alternative Investment Market (AIM). The results portrayed in Fig. 2 show that although the majority of content is unstructured in nature, the sample set had structured and loosely-structured data accounting for 22% to 65% in terms of token count.

Fig. 3 on the other hand shows that although unstructured, free form text and miscellaneous data increased with page count, structured and loosely-structured data could be considered random and unpredictable. The share of structured and loosely-structured information is, in our opinion, substantial and shouldn’t be left out of text mining tasks due to the risk in knowledge loss. We propose syntactically serializing the document in a machine readable format whose schema is flexible enough to adapt to unpredictable content and volatile formatting. It is clear however from this preliminary analysis of the small data set that if it is possible to serialize documents, a format would require several characteristics to make this possible such as flexibility, system interoperability, etc. We review multiple data serialization formats in the next section and select the most appropriate for serializing multifaceted documents written by specialist humans.
A sequence of characters. This data type supports 8, 16 and 32 bit unicode transmission formats.

Either True or False values.

An unordered set of key-value pairs, where the keys must be specified explicitly in a String format.

A vacant value often used as a placeholder.

null

1, -5, 0.9, 34543987584

Either True or False values.

An unordered set of key-value pairs, where the keys must be specified explicitly in a String format.

A vacant value often used as a placeholder.

null

1, -5, 0.9, 34543987584

true or false

false

null
**content oriented format** presents a number of problems for natural language acquisition and analysis. Further, the document’s structure is unpredictable and can change between structured, loosely-structured or free form texts at any time. Therefore, we first propose an introduction stage dedicated to the decomposition of the document, which we call the Acquisition and Segmentation phase of our pipeline shown in Fig 4. Once the pipeline has accepted the document, two concurrent processes must take place. The first process aims to extract the document metadata. This metadata can include information such as creation date, modified date, author name(s), publisher geo-coordinates and other miscellaneous information (13).

We narrow the research direction to seven named entity types, which can be efficiently remodeled using pattern matching techniques such as regular expressions. These entity types are monetary values, percent, decimals, durations, times, dates and miscellaneous quantity values with examples provided in Fig 5.

$$\begin{align*}
&\$1.1b \rightarrow \{\$', 1100000000\} \\
&\$4.7million \rightarrow \{\$', 4700000\} \\
&6.3p \rightarrow \{\GBP', 0.063\} \\
&(12\text{th} \text{March} 2015 \rightarrow \{20150312000000\}) \\
&\€66,000,000 \rightarrow \{\€', 66000000\} \\
&(5.2\% \rightarrow \{0.052\}) \\
&-2\text{perc} \rightarrow \{-0.02\}
\end{align*}$$

Fig. 5. Parsed Tokens: examples depicting an input x and output y where (.) represents an input to the annotator object’s helper function. (x) → y

We advocate the removal of PERCENT objects by converting all percentages found in the text to a decimal format. It was found that performing calculations on the extracted entities was possible if presented in a decimal format, a technique we plan to explore further in future research.

### E. Object Representation

We refer to unstructured content as content that has no predefined data-model and has an unpredictable structure, often mass text that can contain various objects such as money, time, quantities, and so on. We serialize the sentences into an array of JSON objects, maintaining the various indexes discussed in Section A of the pipeline. We use the Punkt Tokenizer in the NLTK as described by Bird (13) which is a model trained using an unsupervised machine learning algorithm for sentence boundary detection. However, various popular tokenizers could also prove appropriate for this task such as the Stanford Tokenizer or TrTok which prove effective when parsing messy web text data. We provide a serialization example of an unstructured paragraph in 5.1.

**Example 5.1 (Unstructured Paragraph Example):**

```json
{
  "sentences": [
    {
      "paragraph_index": 1,
      "sentence_index": 1,
      "cluster_index": 1,
      "page_index": 1,
      "sentence": "This is a sentence."
    },
    {
      "paragraph_index": 1,
      "sentence_index": 2,
      "cluster_index": 1,
      "page_index": 1,
      "sentence": "...also a sentence."
    }
  ]
}
```
Khusro et al (15) note that the detection, extraction and annotation of tables within heterogeneous documents have been quite a significant research problem in Information Retrieval for many years. The structured serialization stage of the hybrid pipeline aims to serialize complex and unpredictable tables into JSON format. Tables are information rich data stores, which contain a lot, often audited factual or objective information. We show how a parsed table (Table II) would be serialized in JSON in Example 5.2. This is possible using a very carefully programmed set of rules. This white box approach has proved effective for our small sample of documents but may need to be extended with intelligent based processing as our architecture scales and formats become increasingly complex.

Example 5.2 (FDPL Table Serialized):

```
Example 5.2 (FDP .L Table Serialized):
{
  "profit_loss": {
    "revenue": {
      "2014-08": 37507000,
      "2013-08": 34381000
    },
    "cost_of_sales": {
      "2014-08": -27606000,
      "2013-08": -25313000
    },
    "gross_profit": {
      "2014-08": 99000000,
      "2013-08": 90680000
    }
  }
}
```

Finally, many messages published over the web are transmitted natively in a JSON format and could prove appropriate to build up a profile of different sources into a single object for efficiency. We believe from this early study that this would make Big Data analysis on these documents much more efficient and manageable (15).

VI. FUTURE WORK

In this initial study we have proposed a novel architecture to serialize and represent a document and its contents for further analysis. We feel there is still scope for vast improvement and research into converting presentation oriented documents into a machine readable format that a human can easily debug. First, with the rise of the semantic web and linked data, we believe that extending our serialization model to JSON-LD (JavaScript Object Notation for Linked Data) may be appropriate. Secondly, there is potential to cluster knowledge together as in Example 6.1 to solve various interoperability and distribution problems within Big Data systems.

Example 6.1 (Multiple Sources Serialized):

```
Example 6.1 (Multiple Sources Serialized):
{
  "id": 1,
  "name": "Company X",
  "collection": {
    "disclosures": { ... },
    "social_media": { ... },
    "company_news": { ... },
    "media_news": { ... },
    "stock_price": { ... }
  }
}
```

REFERENCES


