Using Regulatory Instructions for Information Extraction

Thomas Y. Lee

University of Pennsylvania, The Wharton School
573 JMHH, 3730 Walnut Street, Philadelphia, PA 19104
thomasyl@wharton.upenn.edu

Abstract

In this paper, we describe a novel approach for learning to extract content from the text segments of regulatory filings for the purpose of competitive analysis and regulatory audit. Existing strategies that rely upon an explicit schema or a training set of representative documents are less suited for managing thousands of idiosyncratic submissions by independent filers. We introduce a technique that learns from regulatory instructions. Knowledge about document structure is drawn from the policy documents to initialize a set of extraction patterns. Patterns are relaxed to account for single insertion, deletion, and substitution errors within individual filings. Preliminary results are reported on various sets of filings submitted to the SEC in 2004 and 05.

Introduction

The Government Paperwork Elimination Act, enacted in October 1998, has driven a steady migration from paper to electronic filings in response to regulatory requirements. By many measures, the effort has been a success. For example, through August 2006, the U.S. Securities and Exchange Commission (SEC) has received more than 4.5 billion electronic filings. These filings are read by investors for risk management, by industry partners and competitors for strategic planning, and by Federal auditors for enforcing compliance with public policy.

In addition to numerical figures, regulatory submissions include optional prose that reports information such as "forward-looking" statements and factors affecting market risk (SEC 10-K Item 7 and 7A)(SEC 2005); industry-specific comments such as "risk-factors" (SEC 10-K Item 1A.) compare filings between multiple companies within a single industry to establish sector-specific norms.

Due in part to complexity and volume, comprehensive reviews of electronic filings are not common. For the SEC, even with Sarbanes-Oxley and electronic-filing, mandated three-year audits for every filer may be limited to narrow spot-checks (Leone 2003). Electronic filing is no panacea. Current SEC submission guidelines are limited to text or optional HTML formatting (SEC 2006a). Although XML element definitions are being studied by all government agencies, such proposals do not address the textual elements of regulatory filings (see, for example, XBRL and proposed data elements for Institutional Controls (SEC 2006b).

We introduce a novel approach for automatically extracting text fragments from semistructured regulatory filings. Rather than learning from a manually labeled training set of representative documents, we begin with a single reference source, the regulatory instructions. Using the labels defined in regulatory instructions (we use 'label' in the label-value sense of semistructured data), we manually initialize a set of extraction patterns. The ordering of those labels within a filing is defined in the regulatory requirements and serves as a constraint. A greedy algorithm uses the order constraint to guide the learning of extraction patterns that adjust for the inconsistencies and idiosyncracies within different filings.

The described work is unique in that it learns from regulatory instructions rather than from a training set of exemplar documents. This approach is particularly suited to the regulatory context where submissions are filed independently; submissions may loosely adhere to common instructions yet be riddled with idiosyncrasies due to filer independence or to changes in the regulations.

In the remainder of this paper, we provide some motivating background, describe our approach, detail some preliminary experiments, contrast our approach to related work, and discuss future work.

Motivation

Information extraction (IE) methods enable SQL querying of semistructured documents via wrappers that extract text fragments into relations. However, these techniques typically assume that all documents are generated by a single source and/or are generated from a single template (e.g. product pages from an online retailer or departmental seminar announcements). As a consequence, structural elements such as HTML markup or headings are consistent. If wrapped pages deviate from the norm, the variations are typically few in number and limited to well-understood additions (or omissions).

By contrast, regulatory filings are prepared by independent companies. Despite a common set of instructions, the headings and labels in different submissions may vary in subtle or even glaring ways. Figure 1 contains a portion of the regulatory instructions titled "General Instructions" (GI) for SEC 10-K filings valid from 1/00 through 2/05 with excerpted submissions of three firms from 2005. The text in Figure 1 omits markup to reveal the different ways in which filers deviate from the GI:

- Insertions: the word 'Consolidated' in Item 8.
- Deletions: 'and Supplementary Data' in Item 8.
- Substitutions: 'Disclosure' v. 'Disclosures' and 'of' v. 'About' in Item 7A.
- Transpositions: 'Qualitative and Quantitative' v. 'Quantitative and Qualitative' in Item 7A.

Moreover, because of cross-referencing, a single submission might repeat the same text label numerous times (e.g. 'Item 8. Financial Statements and Supplementary Data'). Thus, the challenge includes not only discovering section headings but also distinguishing the correct instance of a heading.

SEC General Instructions for Form 10-K, last updated 12/05 **Item 7A. Quantitative and Qualitative Disclosures About Market Risk.** Furnish the information required by Item 305 of Regulation S-K (§ 229.305 of this chapter) **Item 8. Financial Statements and Supplementary Data.** Furnish financial statements meeting the requirements of Regulation S-X (§ 210 of this chapter)
2005 Federated Department Stores Inc **Acc No. 0000950152-05-002623** `Item 8. Consolidated Financial Statements and Supplementary Data.`
2005 BERKSHIRE BANCORP INC **Acc No. 0000950117-05-001186** `ITEM 7A. Quantitative and Qualitative Disclosure About Market Risk.`
2005 Cherokee Inc **Acc No. 0001104659-05-016467** `Item 7A. QUALITATIVE AND QUANTITATIVE DISCLOSURES OF MARKET RISK`

Figure 1. Comparing the text of actual 10-K filings to the SEC General Instructions (SEC 2005)

While a human can resolve subtle variations and detect cross-references, automated IE strategies are limited by the positive and negative examples captured in training sets; regulations change over time, exacerbating the issue.

Markup provides little value in this context. Figure 3 depicts filings from the same firms as in Figure 1, but with the original markup included. Not all firms submit using HTML, and those that do, use the markup inconsistently. An additional excerpt in Figure 3 illustrates how even the same firm uses HTML inconsistently over time.

Approach

Our approach is summarized in Figure 2. Documents are normalized by removing all mark-up. We restate the problem of extraction as a search for labels that separate the document into the fragments of interest. Regulatory instructions define a sequence constraint that enforces label ordering. A greedy, hill-climbing algorithm, which is guided by the sequence constraint, incrementally expands the search for robust extraction patterns that apply within a submission that deviates from the GI due to a single insertion, deletion, or substitution error in labels.

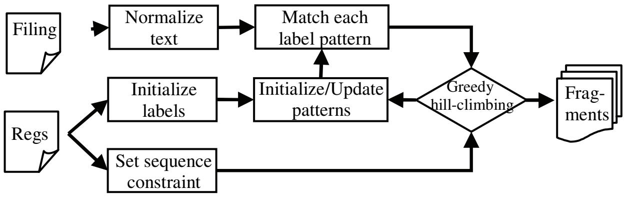

Figure 2. General approach

Document Model

The SEC does not mandate that filing submissions contain markup. Thus, we first normalize filings by removing all SGML-based (e.g. HTML, XML, XBRL) markup. Special-character codes (e.g. Copyright, Registered Trademark) are replaced with corresponding ascii text strings. For purposes of pattern generation, documents are then modeled as a sequence of the following tokens:

- Any mixed-case alphanumeric string literal that appears in the text (abbreviated *i*)
- Punctuation: a non-alphanumeric or white-space (denoted as *p*)
- An unspecified alphanumeric string literal or punctuation (denoted as *S*)
- Any consecutive sequence of white-space tokens included tabs and line-feed (denoted as *s*)
- The symbol + denotes the concatenation of two tokens and _ represents any single token.

A label is a sequence of tokens. A *DocSchema* (which we might think of as a form template), is a sequence of labels that separates a document into a disjoint set of fragments that cover the document.

A filing conforms to the GI if it is an instance of the DocSchema. An instance can be evaluated for conformance by transforming the DocSchema into a non-deterministic finite state machine (NFA) where transitions are represented by labels and states constitute the labeled text fragments. The machine returns *true* if all labels are found in the correct order. Additional transitions could account for optional (or missing) labels/fragments. Non-determinism is required in the presence of cross-references.

Restating the Problem

In the context of a DocSchema, we can restate the problem of extracting text fragments from regulatory filings in two parts. First, for a single submission, we must discover a pattern for each label in the DocSchema. Patterns will differ between submissions due to the idiosyncrasies of individual filers. Within a single submission, a pattern may match multiple times due to cross-referencing. Second, to resolve the cross-references, we must discover the correct match for each pattern.

- *D* is the set of all filings
- *X* is a set of labels
- *Y* is a set of patterns

2005 Federated Department Stores Inc Acc No. 0000950152-05-002623 `<P align="left" style="font-size: 10pt"><B>Item&nbsp;8. Consolidated Financial Statements and Supplementary Data.</B>`
2005 BERKSHIRE BANCORP INC Acc No. 0000950117-05-001186 `ITEM 7A.  Quantitative and Qualitative Disclosure About Market Risk.`
2005 Cherokee Inc Acc No. 0001104659-05-016467 `<p style="font-family:Times New Roman;font-size:10.0pt;font-weight:bold;margin:0pt 0pt 5.0pt 47.0pt;page-break-after:avoid;text-indent:-47.0pt;"><a name="Item7a"><b><font size="2" face="Times New Roman" style="font-size:10.0pt;">Item 7A</font></b></a>.<font size="1" face="Times New Roman" style="font-size:3.0pt;">&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;&#160;</font>QUALITATIVE AND QUANTITATIVE DISCLOSURES OF MARKET RISK</p>`
2004 Federated Department Stores Inc 2004 Acc No. 0000950152-04-002901 `<B><FONT size="2">Item&nbsp;8.&nbsp;Consolidated Financial Statements and Supplementary Data.</FONT></B>`

Figure 3. Comparing the use of HTML markup by different companies and by the same company over time

- Z is the set of integers where z denotes the token offset from the beginning of a filing
- $DocSchema^{*} = [x_i|$ for x_i in $X;\ i = 1..n]$ in 2^X
- $f(X \rightarrow 2^Y)$ maps labels to sets of patterns
- $match(Y,D \rightarrow 2^Z)$ the match set of y in d
- $\mu_i = Union(match(y_i,d))$ *over all* y_i *in* $f(x_i)$
- $g(2^X, 2^Z \rightarrow true, false)$

The *DocSchema* is derived directly from the regulatory instructions. For robustness in the face of filing inconsistencies, each label may correspond to multiple patterns. Thus, we use μ_i to denote the 'match union' or the union of the set of all matches for all patterns corresponding to a single label. Using the notation loosely, a sequence $q = [z_i|z_i$ *in* Z *for* $i = 1..n]$ is drawn from 2^Z where n is the length of the corresponding DocSchema. Our problem is thus reduced to discovering a function f that generates robust patterns from a DocSchema to derive a set of sequences Q, and a function g that takes a DocSchema and a sequence to discover q^*, a sequence in Q that extracts the text fragments specified by the DocSchema.

Generating Robust Patterns

The task of extraction requires identifying the string patterns within a submission that correspond to the labels within a DocSchema. The simplest pattern is to match the raw text of the label as a literal string. In this case, the *raw* function f returns a set containing the single pattern equal to the raw text. Unfortunately, because independent filers deviate from the GI in both simple and in unexpected ways, the simplest pattern is not always reliable.

As a baseline pattern, we use a tokenized version of the label. The *baseline* function f takes as input the raw text of a label and returns a set containing the single pattern of corresponding tokens:

- All string literals are generalized to a mixed-case equivalent i
- All punctuation marks are reduced to an optional token p
- All continuous strings of white-space separators are reduced to a single s

The baseline pattern accounts for the simplest ways in which filers deviate from the GI: inconsistent use of case, white-spacing, and punctuation within labels. Figures 1 and 2 provided examples of more complex ways in which filings deviate. We summarized those deviations as some combination of insertion, deletion, substitution, and transposition errors. In this paper, we focus on deviations due to a single insertion, deletion, or substitution error. Transposition may be modeled as a sequence of insertions and deletions.

In a singleton insertion error, filers introduce an extra literal into a label. In Figure 1, some filers have introduced the string 'Consolidated' into the label for Item 8. The *insertion* function f processes a baseline pattern and returns a set of patterns where each pattern inserts an unspecified literal S into a whitespace. The set of patterns returned by the *insertion* function can be modeled by a transducer. Deletion and substitution errors follow the treatment for insertion errors; we can construct similar transducers for singleton deletion and singleton substitutions. Further details are omitted for space reasons.

Discovering the Correct Pattern Matches

The functions f produce sets of patterns that attempt to compensate for inconsistencies in regulatory filings. However discovering the correct patterns for a particular filing is not enough. Because of cross-references, a single pattern may match multiple times within a document. In our model, labels segment the document into fragments for extraction. If we match a cross-reference rather than the actual label, we will extract fragments incorrectly.

For a given label, we can model the entire, corresponding pattern space as a tree (see Figure 4). The tree is rooted by a label x. From the root, a child is created for each pattern generation function f. Children are ordered by pattern complexity in keeping with a greedy-search strategy (see below). Every function generates one or more child patterns y that in turn match zero or more

[*] We use the notation loosely here to define a sequence from the set of label sets 2^X.

times where a match is identified by the integer *z* giving the offset from the beginning of the file. The search space for a DocSchema is thus the forest of trees, one tree for each label.

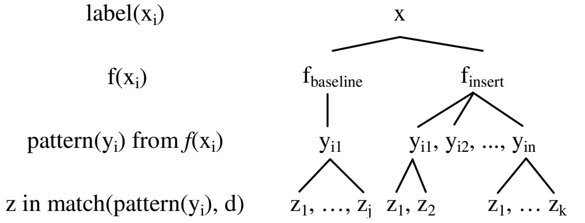

Figure 4. Pattern space as a tree

Our approach for discovering the correct matches within the search space is summarized as a greedy, hill-climbing algorithm that is dictated by the DocSchema's sequence constraint. For an order-constrained sequence of labels, we recursively traverse the label sequence. For each label, we descend the corresponding pattern-tree in a depth-first manner to generate a candidate match *z*. The algorithm is greedy in that it takes the first match that allows forward progress to the next label, "climbing" from label to label.

Preliminary Experiments

In a set of ongoing experiments to evaluate the efficacy of using regulatory instructions to direct the processing of electronic filings, we have focused on the set of SEC 10-K submissions for 2004 and 2005. These filings correspond to a common set of General Instructions (GI) issued in November 2000 and valid through December 2005 (SEC 2005). For this period, the GI defines 23 labels corresponding to 22 fragments of interest. Our analysis is in the earliest stages and so we present only preliminary results.

Problem Significance

Our first question was to assess the significance of the problem. Anecdotally, filings contain inconsistencies; however, the extent of the problem is undocumented. Using the raw text drawn directly from the Regulatory Instruction (RAW), we first noted whether each of 23 labels even appeared within individual documents. RAW matches establish a lower-bound on the significance of the problem because no-match signals an obvious error while the presence of a match does not guarantee accuracy. Because RAW might seem unreasonably strict, we also compared match performance to our *baseline* function (BASE) which adjusts for case-sensitivity, white-spaces, etc. Figure 8 shows the number of times each label pattern matched in a random set of 100 documents.

As expected, in a context where filers have wide latitude in their submissions, the RAW labels match in fewer than 20% of all documents. Moreover, shorter labels (e.g. those with fewer words) match more frequently, consistent with the intuition that longer labels have greater opportunity for error. Furthermore, the BASE performance indicates that the errors vary far more widely than simple issues of white-space, punctuation, and case sensitivity. We conducted repeated trials over random sets of 100 to 150 documents and all performed similarly.

Candidate Generation

Our method for discovering patterns for document segmentation and extraction involves two steps: the generation of candidate patterns and the selection of the correct matching sequence for extraction. In particular, we considered match performance for *insert* (INSERT), and *substitution* (SUB). Performance is benchmarked against (RAW) and (BASE) in Figure 8. Note that in candidate generation, we are only interested in whether the label-pattern matches.

INSERT is nearly indistinguishable from BASE. In

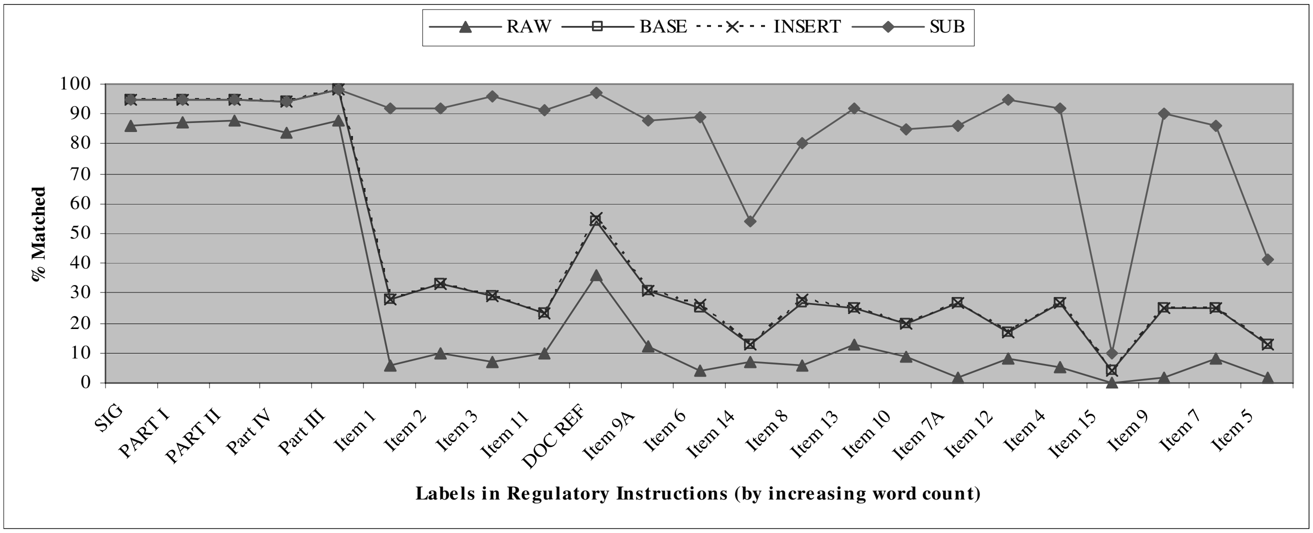

Figure 5. **Matching performance of different pattern generation functions**

retrospect, the similarity between BASE and INSERT is unsurprising given that the pragmatics of grammar permit few opportunities for singleton insertions or punctuation in labels beyond an errant typo. By contrast, despite the restriction to a single deletion or substitution, SUB consistently performs more than twice as well as BASE and INSERT. While Figure 1 illustrates both singleton errors and sequences of errors, these exploratory results suggest that even single-stage corrections can provide a substantial degree of robustness.

As with BASE and RAW, longer labels match less frequently. The label for Item 5 contains eight times the number of words in the shortest labels: "Item 5. Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities." These longer labels are where we would expect to benefit the most from more robust pattern generation. Additional trials on sets of 12,000 randomly selected submissions from 2004 and 2005 showed a slight decline in the matching percentages, but SUB continued to match above 85% for nearly all labels.

Candidate Selection

It is not enough to simply discover matches in the text, however. For purposes of extraction, the underlying task is to select the correct sequence of matches. To evaluate the constraint-driven selection of label matches, we measure the precision and recall of extracting the 22 fragments delimited by the 23 labels in the GI. In our context, text is extracted by segmenting a regulatory submission on the labels and match points from our algorithm. Precision measures the number of extracted fragments that correctly match the actual text. Recall measures the number of real fragments that we correctly extracted. Note that, despite the instructions, many filings may omit one or more filing sections. As a consequence, the denominator for recall is not simply the number of documents in the test set. Moreover, the extraction of a segment can fail in at least two ways. We might select the wrong separator label at the *beginning* of a segment, or we might incorrectly identify the label text at the *end* of a segment.

This stage of our evaluation is too preliminary for definitive conclusions. However, to illustrate the on-going analysis, Table 1 provides the precision (P) and recall (R) for a sample of eighteen randomly selected documents. The relatively high precision and recall, even for longer labels with fewer matches, suggests some promise to the approach. More comparative analysis is warranted.

Table 1. Precision and recall for extracting text fragments

	PART I	Item 1	Item 2	Item 3	Item 4	PART II	Item 5
P	0.35	0.88	1.00	0.76	0.92	0.94	0.71
R	0.35	0.83	0.33	0.72	0.67	0.88	0.67
	Item 6	Item 7	Item7A	Item 8	Item 9	Item 9A	Part III
P	0.92	1.00	1.00	1.00	0.94	1.00	0.82
R	0.61	0.88	0.94	0.94	0.94	0.94	0.78
	Item 10	Item 11	Item 12	Item 13	Item 14	Part IV	Item 15
P	0.87	1.00	0.19	0.88	0.94	0.94	1.00
R	0.72	0.24	0.19	1.00	0.94	0.94	0.94

Related Work

The work described in this paper draws from two different research streams: Structured Information Retrieval (IR) and Information Extraction (IE). While borrowing from both streams, regulatory documents introduce a new, previously un-addressed dimension to past challenges.

Borrowing directly from the structured IR community, we model documents as a flat object comprised of an ordered sequence of disjoint fragments covering a document (Baeza-Yates and Navarro 1996). Structured IR exploits the intuition that knowing *where* terms appear within documents can affect relevance. In the case of regulatory filings, it is difficult to attribute a search term to a particular document fragment if the labels separating fragments cannot be accurately identified. By identifying label patterns for individual documents, we hope to facilitate the structural indexing of content that previously lacked relevant cues.

While learning segment labels may complement structured IR, the IE community focuses on extraction delimiters of which our labels are an example. Research in learning IE varies along at least two dimensions: the degree of machine-learning supervision and the exploitation of document structure, varying from highly-structured mark-up to the uncertainty of English (language) grammar rules (Soderland 1997; Banko et al. 2002).

The majority of the work in IE adopts a supervised, machine learning approach. A hand-coded set of positive and negative examples is generated from a representative set of documents. In either a top-down (Soderland 1997; Freitag 1998) or bottom-up (Califf and Mooney 1999) fashion, a set of disjunctive rules is generated to cover the greatest number of positive labels while excluding negative instances. Refinements draw upon context such as where items appear relative to a specified label or relative to one-another (Kushmerick, Weld and Doorenbos 1997; Muslea, Minton and Knoblock 2001).

Our approach is most similar to those who use transducers to model document context (Hsu and Chang 1999). The differences in our work stem from the unique challenges of managing submissions from thousands of independent filers. Rather than a training set of instances, we use policy documents to manually identify the relevant labels and label ordering.

Second, context elements are used in the prior literature to identify explicit regions of text within which a separate extraction process might take place. Instead, we use context (sequences) to resolve the ambiguity that can arise from cross-references in the text.

Future Work

While the field of Information Extraction (IE) has been a popular subject for research, the management of regulatory filings introduces new challenges. Traditional strategies for IE-learning have typically (sometimes implicitly) assumed a substantial degree of consistency in the explicit structure of the source documents. Typical IE content, including comparison guides, product or service reviews, classified ads, and seminar announcements, is generated by a single provider.

By contrast, the government processes regulatory submissions from thousands of unique filers. Though filers begin with a common form, no two filings are alike. The General Instructions (GI) for SEC Form 10-K explicitly note that they are "not to be used as a blank form to be filled in, but only as a guide in the preparation of the report on paper meeting the requirements ….(SEC 2005)" In terms of markup, even if filers agreed on a set of items to tag, each filer could create their own tag names. Moreover, XML (XBRL in the case of the SEC) does not necessarily promise any resolution to the problem of managing the *text* portions of filings.

In this paper, we have introduced a technique for discovering extraction patterns based upon a reference document. From the reference document, we initialize a set of candidate patterns (the baseline), and a constraint on the order of those patterns. The sequence constraint directs a greedy strategy for discovering the correct fragmentation of a document into its constituent parts. We tested the algorithm using the SEC GI for Form 10-K as a reference document to process submissions from 2004 and 2005.

While the work presented here is still preliminary, it does provide a guide for future work in two directions: candidate generation and candidate selection. For candidate generation, our current approach allows for only singleton errors. However, we deliberately framed the problem in terms of insertions and deletions to highlight parallels to the work on sequential patterns (Agarwal and Srikant 1995) including work on edit-distances.

To select among possible label matches, our current greedy-hill climbing approach embeds an implicit cost-function that selects simpler pattern-generating functions and simpler patterns first. In addition to formalizing this model, we wish to learn fragment characteristics including relative length and relative offset. In the manner of earlier wrapper maintenance systems (Kushmerick 1999; Lerman and Minton 2000), fragment characteristics can be used in candidate selection to detect mismatches.

More generally, this work highlights the similarity between structured Information Retrieval (IR) and semistructured data processing. Reference documents like regulatory instructions serve as an approximate schema both for weighting search terms (IR) and for schema alignment because regulations (and their attendant instructions) change over time. In our work, we are combining these methods to explore a more detailed segmentation not only for regulatory filings but other documents with common specifications such as government contracts.

Acknowledgements

The author would like to thank the University of Pennsylvania Research Foundation, the Wharton eBusiness Initiative, and the Fishman-Davidson Center for their support.

References

Agarwal, R. and Srikant, R. 1995. Mining Sequential Patterns. *ICDE.*

Baeza-Yates, R. and Navarro, G. 1996. Integrating Contents and Structure in Text Retrieval. *SIGMOD Record* **25**(1): 67-79.

Banko, M., Brill, E., Dumais, S. and Lin, J. 2002. AskMSR: Question Answering Using the Worldwide Web. *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Califf, M. E. and Mooney, R. J. 1999. Relational Learning of Pattern-Match Rules for Information Extraction. *AAAI.*

Freitag, D. 1998. Information Extraction from HTML: Application of a General Machine Learning Approach. *AAAI, IAAI.*.

Hsu, C.-N. and Chang, C.-C. 1999. Finite-State Transducers for Semi-Structured Text Mining. *IJCAI Workshop on Text Mining: Foundations, Techniques and Application*.

Kushmerick, N. 1999. Regression testing for wrapper maintenance. *AAAI.*

Kushmerick, N., Weld, D. S. and Doorenbos, R. 1997. Wrapper Induction for Information Extraction. *IJCAI.*

Leone, M. 2003. RX for Fraud: More SEC Checkups. *CFO.com.*

Lerman, K. and Minton, S. 2000. Learning the Common Structure of Data. *AAAI.*

Muslea, I., Minton, S. and Knoblock, C. A. 2001. Hierarchical Wrapper Induction for Semistructured Information Sources. *Journal of Autonomous Agents and Multi-Agent Systems* **4**: 93-114.

Securities and Exchange Commission, U.S. 2005. Annual Report Pursuant to Section 13 or 15(d) (Form 10-K) General Instructions.

Securities and Exchange Commission, U.S. 2006a, 2/6/2006. EDGAR Filer Manual (Volume II). 3. 2006.

Securities and Exchange Commission, U.S. 2006b, 05/01/06. FAQ: XBRL Voluntary Filing Program.

Soderland, S. 1997. Learning to Extract Text-based Information from the World Wide Web. *KDD*.