2013 International Conference on Advances in ICT for Emerging Regions (ICTer): 110 - 117

Ontology Based Annotation Mechanism for Financial Documents

Perera, K[#*1], Karunarathne, D.D.[*2], Siriwardena, A.[#3], Balaretnaraja, D.[#4]

[#]*Lanka Software Foundation*
Colombo, Sri Lanka
[1]kasunp@opensource.lk
[3]amal@opensource.lk
[4]sdbala@gmail.com

[*] *University of Colombo School of Computing*
Colombo, Sri Lanka
[2]ddk@ucsc.cmb.ac.lk

Abstract— **The Vast number of publicly available electronic financial documents and document repositories and their rapid growth pose a great challenge in understanding, managing and structuring the information. Due to several reasons content of these documents is open to variety of differing interpretations and resulting ambiguity. Annotating these data with semantics to constrain the inconsistent interpretation of data facilitates better reuse and interoperability. We propose a semi-supervised approach for creating annotations for the extracted text of financial documents. A Supervised approach would include human experts in the annotation process. Unsupervised or machine based annotation is done by recommending Financial Industry Business Ontology (FIBO) terms for document sections based on the Okapi Similarity measure. Annotation data can be used to infer knowledge from important sentences or document sections to gain better understanding or decision making. Our annotation results indicate that similar pairs of sections have more common FIBO terms and different pairs of sections have a lesser number of similar FIBO terms.**

Keywords— **Ontology, Annotation, Semantic Web, Financial Data**

I. INTRODUCTION

Today a vast volume of financial/ business documents such as company fillings, bond prospectuses, master agreements, fixed deposit documents, credit card agreements etc are available for the public on the internet. These documents tend to be distributed over multiple web sites/repositories, such as Electronic Municipal Market Access system (EMMA) [1] the municipal bond repository, Consumer Financial Protection Bureau (CFPB) [2] credit card database, U.S. Securities and Exchange Commission (SEC)[3] filings repository etc.. The type of these documents can be categorized as unstructured, semi-structured by using document formatting elements, semi-structured by using pre-defined tags such as html, XML, csv documents and structured as tables by using some predefined classification/s .

The information in these documents contains a wide variety of information such as financial and legal contracts, descriptions of financial products etc.. Some of these documents are based on standard templates. However different organizations may have introduced minor or major modifications to the base templates and also the base templates may have been subjected to minor and major variations over time. For example, companies might have used completely different sub-headings for similar-content sections, or may have moved some sections from the body into Appendices, or may have split the same content into sub-sections with separ headings etc.

Since the documents have originated from different source and been compiled to maintain the consistency within the loc domain during a specific time frame, they are open to a varie of differing interpretations and resulting ambiguity [4]. Henc the rapid growth of publicly available electronic documents an document repositories pose a great challenge in understanding managing and structuring the information.

Our current work is motivated by the need for tools that ca improve the accuracy of the data interpretation enablin consistent integration of documents in the financial domai These tools would contribute to making the Semantic web reality by annotating data with semantics to constrain th inconsistent interpretation of data and hence facilitate bett reuse and interoperability [5].

Applying computational technologies, open standards an ontologies to address difficult modeling and structuring c publicly available financial data, would result in improved too for regulators to monitor financial systems as well as explor the hidden wisdom buried in financial contracts and document Document annotation with the aid of ontologies woul effectively structure this unstructured information and offe easier access to the knowledge in the documents for bot human users and computers.

In this paper, we focus on business data sources in th financial domain, with particular emphasis on the bon prospectus of The Electronic Municipal Market Access syste (EMMA) managed by The Municipal Securities Rulemakin Board (MSRB). EMMA is a comprehensive, centralized on-lin source providing free access to municipal disclosure municipal bonds. These bonds are issued by states, counties cities or their agencies to finance public-purpose projects schools, roads, bridges, utilities, affordable housing, airport hospitals, and other public facilities and programs. The term defined in the Financial Industry Business Ontology [6] woul be used the markup/annotate financial documents and would b saved as metadata. Annotation would provide information o important sentences or document sections to enable bett understanding or decision making

978-1-4799-1276-6/13/$31.00 ©2013 IEEE

We would extract financial concepts from bond documents annotate using the FIBO terms. These bond documents are ructured (text-PDF), thus to extract financial instruments or concepts and integrate with ontology terms is a crucial part the annotation process. This annotated and structured rmation of unstructured bond documents would be of rest to regulators, investors, financial analysts and bankers researchers in the financial domain.

he paper is structured as follows: section 2 reviews related ects in the field, section 3 discusses the background and nologies that we have used, section 4 provides the details of approach, section 5 presents an evaluation framework and lts of the experiment, section 6 discusses the results and ines directions for future work.

II. Related Work

he development of Semantic annotation tools came to light the last couple of years with the development of the antic Web.

One of the most popular and widely used algorithms for ument retrieval and document similarity calculations tasks ne Term Frequency- Inverse Document Frequency (TF-IDF) ghting mechanism. The term frequency (TF) is the number imes that a term t appears in a document, i.e. raw frequency the term in the same document. The inverse document uency (IDF) is calculated by dividing the total number of uments by the number of documents containing the term, then taking the logarithm of that quotient, thus indicating ther the term is common or rare across all documents

n this review we list some of these tools that are related to logy derived semantic annotation.

Midas project at IBM [7],[8] uses information extraction, rmation integration, and scalable infrastructure to estigate behavior of financial institutions either at the whole tem level (i.e., systemic analysis) or at the individual pany level by focusing on high quality financial data rces such as U.S. Securities and Exchange Commission C) filings and Federal Deposit Insurance Corporation DIC) fillings.

Midas takes input data such as SEC and FDIC fillings and duces the output as sets of integrated and cleansed objects l relationships between those objects. Midas workflow olves five main steps:

Crawl - retrieve data directly from public data sources and ring it in local file system.

Extract- annotate unstructured data using the SystemT, a rule-ed (AQL rule language) information extraction system. 3-ity Resolution- identifies and links annotated objects to the responding real-world entity.

Map & Fuse- transforms annotated data into a set of objects l relationships between those objects.

At the final step it creates entities and a relation index using ll known text search engine, Lucene.

SemTag [9] is an application that performs automated antic tagging of large corpora of web pages using the TAP owledge base as the standard ontology. TAP contains lexical l taxonomic information about music, movies, authors, orts, autos, health, and other popular objects. SemTag is veloped on a Seeker platform which supports sharing the semantic tags with other applications, large scale text analysis and provides core annotation algorithms. SemTag has been tested on approximately 264 million web pages, and has generated approximately 434 million automatically disambiguated semantic tags. Users have identified that resolving ambiguities in a natural language corpus is a key challenge in the SemTag tool and they have introduced a new algorithm called TBD (Taxonomy-Based Disambiguation) for disambiguation on text corpus. SemTag works in three passes;
1-Spotting pass: documents are retrieved, tokenized, and then processed to find instances of approximately 72K labels of TAP knowledge base.
2-Learning pass: sample of data is scanned to determine distribution of terms at each internal node of the taxonomy.
3-Tagging pass: each reference is disambiguated and a record is inserted into a database.

The KIM (Knowledge and Information Management) platform [10], [11] is a tool for automatic ontology-based named entities annotation, indexing and retrieval based on GATE (General Architecture for Text Engineering) developed by University of Sheffield. The KIM presents a semantic annotation model, consisting of lightweight ontology (KIMO), a semantic repository, and a metadata storage model. It has mainly focused on the automatic semantic annotation, leaving out manual annotation. KIM is build on top of GATE architecture and has used GATE's text processing functionalities, such as tokenization, splitting to sentences and part-of-speech tagging, extensively. A semantic gazetteer is used to generate lookup annotations. Ontology aware pattern-matching grammars are used to recognize named entities with class and instance information, referring to the KIM ontology and the knowledge base. Later, simple disambiguation techniques are used to solve the ambiguities within the named entities. Based on the recognized semantic constructions, template relation construction is performed by means of the grammar rules. As a result, the knowledge base is enriched with the recognized relations between entities. Previously unknown aliases and entities are added to the knowledge base to finalizes the IE process, having as a result named entity annotations linked to their semantic descriptions in the KB.

Orank is a tool for ranking HTML documents based on an ontology. [12] propose an approach which combines conceptual, statistical and linguistic methods for semantic annotation of HTML pages. They have extensively used natural language processing techniques for extracting phrases and stemming words and an ontology based conceptual method to annotate web documents. Orank's main functional modules are: 1-Ontologyprocessor: assigns weights to the relations in the reference ontology.
2-Query processor: extracts the input query phrases, and then applies weighted ontology to expand these phrases with their related concepts.
3-Document processor: documents are annotated using the ontology and annotation vector is created for each HTML document.
4-Ranker: calculates the rank of each document according to its relevance to the expanded user query.

IdentiFinder is a hidden Markov model that learns to recognize and classify names, dates, times and numerical quantities [13]. It deviates from traditional approaches of data extraction by using a probabilistic language model (a statistical

bigram language model) to handle text robustly, to achieve high accuracy, and to ease maintenance. It has used Wall Street Journal text produced in the 1990s as training and testing data which has been shown to achieve F-measure scores above 90%.

The DARPA Machine Reading program created a corpus of general text readability containing various forms of human and machine generated texts [14]. The aim of this program is to transform natural language texts into a format suitable for automatic processing by machines and to filter out poorly written documents based on the text quality. DARPA Machine Reading program has used a semi-automatic approach to create annotations for the text where it is trained using human annotated text and trained to answer questions or perform some reasoning task

All of the above tools use natural language techniques such as tokenization and part-of-speech tagging to process the text and identify annotation candidates or use rule based information extraction. Our approach fundamentally differs from these tools since natural language processing is out of scope in our tool which makes our tool lightweight and not computationally intensive. Some of the above tools introduce an automatic approach whereas we propose a semi-automated approach putting the human annotator at the center of the annotation process.

III. Background and Technologies

A. Current Representation of Financial Documents

Financial documents in EMMA repository do not contain any semantic information related to its content. These documents are composed with many terms and keywords and important sentences are spread out through the document. The only structured information on these documents would be the section or sub-heading names saved as bookmarks.

The content of the document would be extracted and saved in the annotation tools' database document section wise. In the later stage we would index these extracted documents using Lucene. The stop words, which occur too frequently or are unimportance to the content would be removed during the indexing process. For similarity calculation purposes we consider, sections of EMMA documents as a base documents.

B. FIBO and Definitions

The Financial Industrial Business Ontology (FIBO) defines financial industry terms, definitions, and synonyms using modern semantic web technologies and widely-adopted Object management Group (OMG) modeling standards. FIBO was developed by the Enterprise Data management Group for the benefit of researchers, banks, brokers, vendors and other possible users in the financial industry. FIBO can be effectively utilized in the areas of regulatory governance, data standardization, risk management, data analytics requirements etc.

The FIBO ontology can be viewed from two angles; as a business conceptual ontology and as an operational ontology. The business conceptual ontology defines how the business concepts are interconnected and the conceptual abstractions that they were derived from. The operational ontology can be used developing software applications and tools on the basis of FIBO Ontology. Current online FIBO ontology is organized as a flat structure, which means that the relationships betwe FIBO terms cannot be computationally mapped. T development in this ontology in OWL (The Web Ontolog Language) format is under construction. The ontology tern and definitions are stored using a conventional database tab in order to make it computationally convenient and accessibl For similarity calculation purposes, virtual documents would b constructed for each FIBO term, the content of the virtu document would be the definition of the same FIBO ter These documents are considered as query documents which ar used in similarity calculation against document sections (bas documents) defined above.

These was no other complete financial ontology availab until now, so FIBO will help to connect concepts and data in clear and visible way and the ultimately outcome will be better financial system. We have mainly focused on the FI for document annotations since it provides an excellent se definitions for each FIBO term. But the inability computationally map the relationships between FIBO terms major drawback.

C. The Two Similarity measures

Cosine similarity measure and Okapi similarity measure a two methods commonly used in information retrieval purpose However a l large gap between documents lengths wou make some of the similarity measures inefficient, from whic the need for identifying best similarity measures for long ba document versus short query documents arises.

1) *Cosine Score*: A Given two vectors of attributes, A a B, the cosine similarity, θ, is represented using a dot produ and magnitude as

$$Similarity = \cos\theta = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} * \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

The resulting similarity ranges from −1 (meaning exact opposite), to 1 (meaning exactly the same), with 0 usua indicating independence, and in-between values indicati intermediate similarity or dissimilarity.

For text matching, the attribute vectors A and B are usually t term frequency vectors of the documents. The cosine similari can be seen as a method of normalizing document length duri comparison.

In the case of information retrieval, the cosine similarity two documents will range from 0 to 1, since the ter frequencies (tf-idf weights) cannot be negative. The ang between two term frequency vectors cannot be greater than 90

2) *Okapi Score:* A Given a query **Q**, containing keywor $q_1, q_2 \ldots q_n$, the Okapi score of a document **D** is:

$$Score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * \left(1 - b + b * \frac{|D|}{aveDocLength}\right)}.$$

where $\mathbf{f(q_i,D)}$ is $\mathbf{q_i}$'s term frequency in the document **D**, |D| the length of the document **D** in words, and **avgdl** is t average document length in the text collection from whic documents are drawn. $\mathbf{K_1}$ and **b** are free parameters, usual chosen, in the absence of an advanced optimization,

(1.2-2) and **b=0.75**. **IDF(q_i)** is the IDF (inverse document ıency) weight of the query term q_i is usually computed as:

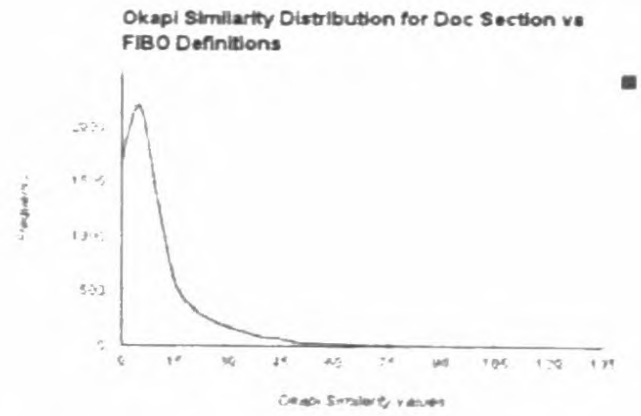
$$IDF(q_i) = log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right)$$

V is the total number of documents in the collection and $n(q_i)$ ıe number of documents containing q_i query term.

) *Identifying a suitable measure:* Reference [15] have stigated behavior of similarity measures on short queries g two methods, Cosine similarity measure and Okapi ilarity measure are two measures that would give good ormance in long queries.

rage length of a base document (document section) was r removing common words) 2255 words per document and average length of a query document (FIBO definition) was r removing common words) 10 words per document. The between the length of a base document and query ıment is at a considerably higher value.

the short queries [15] have found that cosine similarity sure performs badly but Okapi similarity measure gives d performance. For the short query documents (average ds per query document is 10) the test results from our riments, where cosine measure gave very small similarity es and okapi gave acceptable similarity values, agree with ıbove conclusion. Figure 1 shows the distribution of cosine larity results of our experiments; almost every similarity e between the document and FIBO term definition was less 0.3 which is a very low value in the cosine range 0.0-1.0. indicates that almost all of the FIBO term definitions are different to the document content and no significant ber of FIBO terms can be recommended. Okapi score gave ively higher similarity values (Figure 2), and hence Okapi larity measure was chosen as the similarity calculation sure.



g 1 Okapi Similarity Distribution

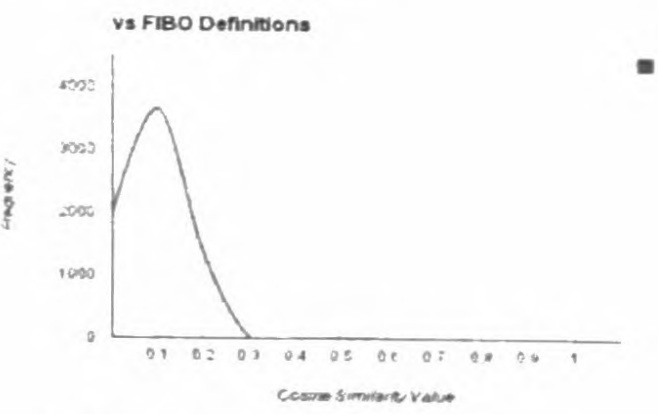


Fig.2 Cosine Similarity Distribution

IV. Methodology and Approach

A financial contract or bond prospectus might be complicated and difficult to understand since financial terms might not be standardized; hence it would be an even more tedious task for a party that lacks a good knowledge of financial terms and definitions. So a buyer and seller may sign important documents without even reviewing them in detail or comparing with similar documents.

Traditional plain text representation of data with structured data in the form of annotations, tags etc. would be beneficial for various document related tasks such as understanding documents, browsing documents, comparing documents etc. [16]. More importantly conceptual representation of text can be used for minimizing the semantic gap between text and user queries in tasks such as search and information retrieval [17].

Researchers have explored different directions of extracting text from financial contracts to create annotations for the documents. Our approach is to use a combination of supervised and unsupervised approach for training the system on document annotation. In general, we are not dealing with the general data extraction or natural language processing for creation annotations but are focusing on the problem of creating annotations using available ontologies and linking the ontology terms with text or specific sentences of the text.

A. Implementation

Our tool adopted the annotation workflow that [18] have proposed which consists of three passes.

1. Lightweight parsing and semantic markup of basic entities and language structures
2. Externalization of the facts to database, which can be then used by search engine for queries.
3. Transformation rules use inferences and structural patterns to infer semantic markup of design facts

However, the last step of the above method is not implemented in our tool, but we wish to work on this step to improve the quality of the semi-automatic annotation. The current implementation is based on the most popular open source full-text search engine-Lucene.

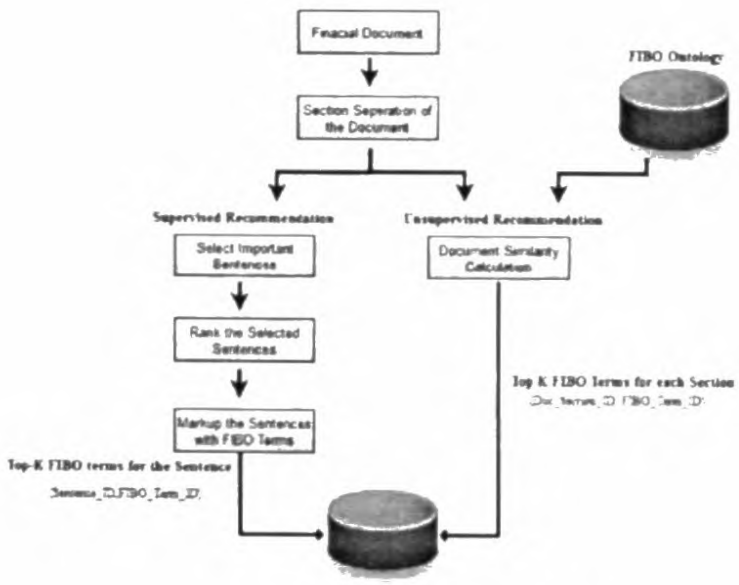


Fig. 3 Annotation Workflow

B. Supervised Approach to Annotate

Human experts would do a semantic annotation of financial contracts at the sentence level. Sentence/ span of text would be connected with a FIBO ontology term/s, where annotation information will be saved into a database. At the end of manual annotation, the target text would transfer into an entity, which document could be described using FIBO terms and definitions. In addition, the sentences would be given a score, where their score would be used to prioritize the sentences in document processing tasks.

As in the figure 3, the system would use a semi-automated approach to separate a bond document into sections. The user would be given an interface where he/she can browse the content of the document section-wise (Figure 4). As indicated in "Supervised Recommendation" path in the figure 1, user would select important sentences from each of the document sections. Then user would rank the importance of the sentences using tags "Very Important", "Important" and "Average" according to the user perspective (Figure 4). Later user would annotate each selected sentence with FIBO term/s and data would be saved to a relational database (Figure 5).

Fig. 4 Screenshot of Web GUI: Selecting sentences and Mark the importance

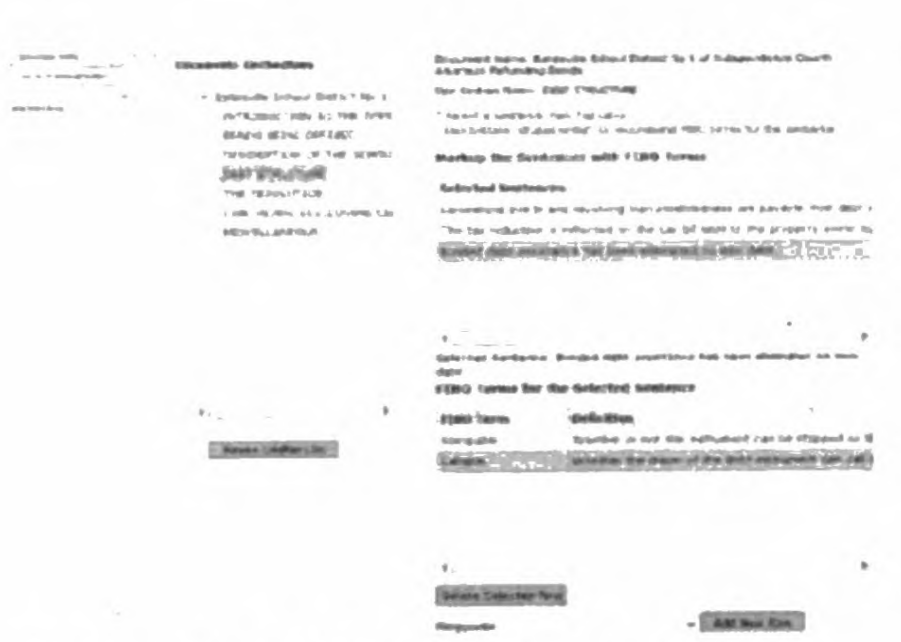


Fig. 5 Screenshot of Web GUI: Markup Sentences with FIBO terms

C. Unsupervised Approach to Annotate

In the "Unsupervised Recommendation" pa recommendation of FIBO terms for documents sections wou be done at document section level but not at the sentence lev as supervised recommendation. First a virtual document wou be constructed for every FIBO term using the term definitio Then the similarity between the sections of documents vs ea FIBO term document (virtual document) would be calculate The FIBO terms that have the highest similarity score to particular document section would be picked as top recommended terms for that document section. Simila annotation data would be saved into a relational database.

We have used a modified version of the Okapi Similar equation, since our main intention to use it for docu similarity, but not for document retrieval only. If two docu doc A and doc B are used for similarity calculation, it sho adhere to the symmetric property i.e Similarity_Score(Doc Doc B)== Similarity_Score(Doc B, Doc A). The general Ok equation could not satisfy this; hence we have used follow version of the Okapi Similarity measure that satisfies symmetric property.

$$IDF(D,i) = log\left(\frac{TotalDocs - docFreq(i) + 0.5}{docFreq(i) + 0.5}\right)$$

$$TFWeight(D,i) = \frac{2.2 * TF(i)}{1.2 * \left(0.25 + \frac{0.75 * docLength(D)}{aveDocLength}\right) + TF(}$$

$$OkapiSim(D,Q) = \sum TFWeight(Q,i) * TFWeight(D,i) * IDF$$

There can be situations where the value of the ***IDF(*** becomes negative (<0) (e.g. equation has to take logarithm value less than 1(<1)). Further it may lead to situations wh final ***OkapiSim(D,Q)*** becomes negative.

Hence the condition below was included to Okapi score avoid the Okapi score becoming negative.

if IDF(D,i) <0 then IDF(D,i) =0

Sections of documents (Base documents) and FIBO t documents (Query documents) are indexed using Lucene, w the configuration for calculating term frequencies of documents. Pseudocode representation of the Simila calculation algorithm is presented in Appendix A.

D. Comparison of document sections based on recommende terms

The annotation tool provides the facility to comp recommended FIBO terms across sections of the document would categorize the document sections based on the sec topic and list the FIBO terms recommended for each sect Users can select a FIBO term from any section and ch whether that FIBO term appears in other sections in a sim nature and over the different sections of the documents. Als has the facility to compare FIBO terms recommended document wise.

Figure 6 shows a snapshot of the web tool interface. table rows show the sections of documents. The grid relate the "The Bonds" section is expanded and shows recommen

)O terms for each document section. When user selects any n from the grid, it will highlight the similar terms across the ;uments.

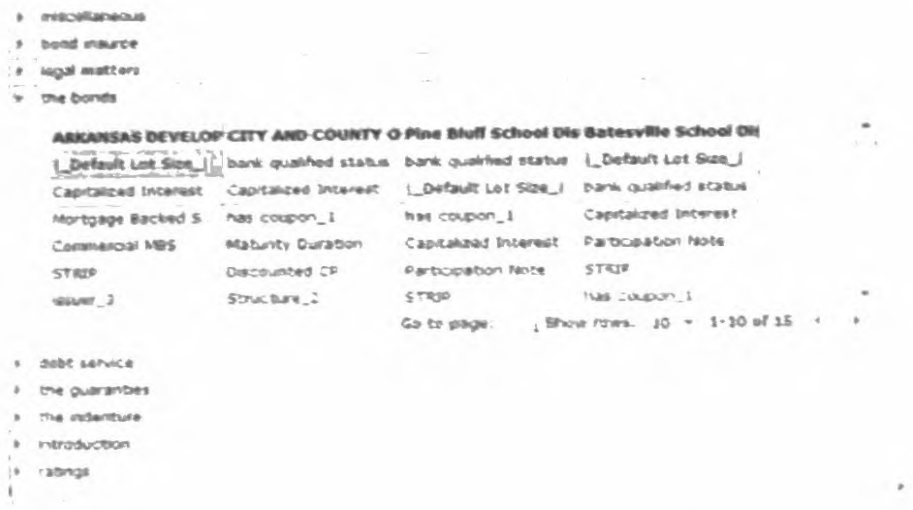


Fig. 6 FIBO term comparison across document sections.

V. EVALUATION AND RESULT

_itial results show that promising FIBO terms are matically recommended for EMMA document sections. he most effective method of calculating the precision and ll of the proposed system would be to compare the system ut directly with the financial domain expert human analysis. vever this type of analysis cannot be applied on a large us since human experts' annotation of a large volume of is not affordable. So choice of an evaluation method for fying the quality is a problem.

Ve conducted a comparison of recommended FIBO terms similar types of document sections across the documents comparing FIBO terms of bond sections of documents) comparison of recommended FIBO terms for dissimilar ions (e.g. Comparing FIBO terms of a bond section with r type of sections)

irst we manually look at the sections of the documents and ter the similar sections based on the given section topic. common sections we identified across documents are given w;
) Introduction, 2)The Bonds , 3)Legal Matters, 4)Security Bonds, 5)Continuing Disclosure, 6)Tax matters , 7)Debt cture, 8)Sources and uses of funds.

or each pair of document sections within the cluster we ulate the number of similar recommended FIBO terms veen two sections as a percentage. i.e. No of similar terms in the pair/ (Total no of terms of pair/2)

For a pair of sections, the maximal similarity of 1.00 is eved if and only if both annotation sets have the same inality (No of terms for section 1 = No of terms for section nd all annotation terms from one section are equal to the r section annotation terms. 0.00 means that the pair does have any common FIBO terms. The average similar score the cluster would be calculated by summing up similarity es for all the pairs and dividing by the number of pairs.

Then we calculate FIBO term similarity for each document section of a cluster with all other documents of different clusters by using the same equation as above. Again we calculate the average FIBO term similarity as above.

Figure 7 indicates the results of our experiment.

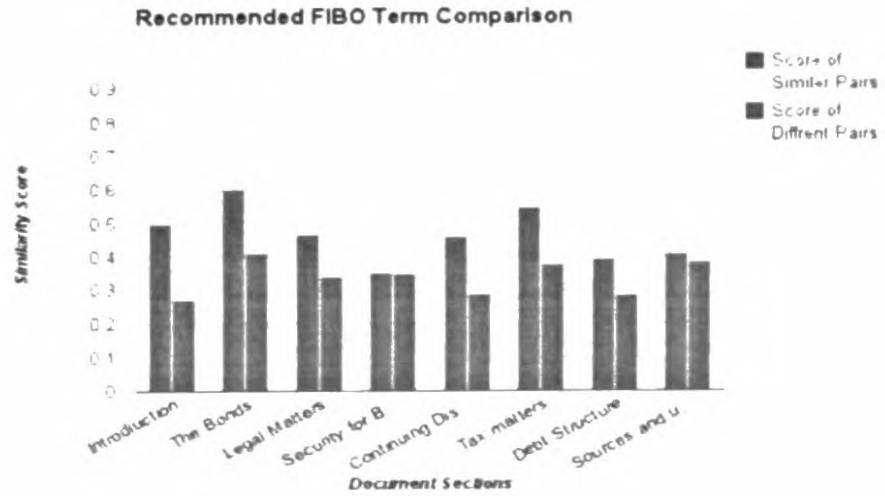


Fig. 7 FIBO term similarity comparison

Results clearly indicate that similar pairs of sections have more common FIBO terms and different pairs of sections have a lesser number of similar FIBO terms. This indicates that the system recommends similar FIBO terms for similar sections.

On the 1 to 0 scale system recommended terms for similar sections have an average of 0.465 similarity, whereas terms recommended for non-similar sections have an average of 0.338 common FIBO terms. By rounding the values we can conclude that the system recommends FIBO terms for similar sections with a probability of 0.5, i.e. at least half of the FIBO terms are common across similar sections.

Security for bonds section pairs indicates a lower similarity value compared to other sections. A possible reason would be Security for Bonds sections having less content compared to other sections.

The system can be also used as a tool to increase the productivity of human annotators. Manual annotation of documents would consume more time since human experts need to read the document word by word. When a human expert works directly on the original document he/she can make errors or miss some items because of lack of attention; when working on the document already annotated by the tool he/she can easily note the defects and therefore produce a higher quality annotation. Since documents are automatically separated through the system, human can leave out the unnecessary sections easily without reading through the section.

VI. DISCUSSION AND FURTHER WORK

This paper presents a semi-automatic semantic annotation tool that uses textual similarity between ontology terms and financial documents to annotate. This tool is focused on financial data sources where semantic annotation would help to improve the lack of semantic information.

There are several limitations to our current approach to compare similarity of sections/ documents. One is that we only consider identical terms; however, there may be FIBO annotations that are similar but not identical. A second limitation is that we ignore all annotations that do not match; such annotations may signal that some parts of the sections are not at all similar.

AnnSim[19] and AnnSim+[20] are two metrics that model the problem of comparing the annotations of a pair of documents as a bipartite match. AnnSim is a metric for a 1-to-1 maximal weighted bipartite match (MWBM) and AnnSim+ is a metric for a many-to-many MWBM.

In future work, we will explore these solutions and apply them to our problem of comparing the FIBO annotations of financial contracts.

Below are the possible use cases and areas where this application can be further developed.

The semi-automatic approach we have introduced for retrieving information from financial documents can be identified as a hybrid approach of semantic information retrieval and standard text retrieval. This approach includes Semantic Web features like semantic markups, Markup/text relationships and Ontology mapping and also the standard Information Retrieval approaches such as IDF_TF weighting, term indexing, similarity measuring. For building a semantic search engine, Hybrid Information Retrieval is a clever combination for semantic oriented search approaches with standard text retrieval techniques. In this model. the document is also represented in a vector space model but in addition semantic markup or a pair of markup-terms can be defined as a term as well as words occurring in a document. When searching using a query term, markup similarity allows us to rank the result together with text term similarity. Figuring out ontology terms that are most relevant to the query concept could be used to reduce search space.

Another idea would be to cluster the documents in the repository based on the FIBO terms similarity of the documents. This would cluster documents not on the text similarity but on the semantic similarity of the documents which can be used to gain better understanding of the repository.

Further, semantic similarity of two documents from the same organization or different organizations can be investigated based on the pair wise comparison of the documents. System may annotate two document sections with similar FIBO terms but the real document text may contain gaps or dissimilarity.

APPENDIX A- PSEUDOCODE REPRESENTATION OF THE SIMILARITY CALCULATION ALGORITHM.

```
computeOkapiSimilarity(BaseDocumentID, ListOFQueryDocumentID)
Declare a HashMap variable OkapiValue[QueryDocumentID, SimilarityValue]
Declare a HashMap variable DocumentTermFrequencyMap[Document ID,
Map of Terms of the Document]
Declare a HashMap variable IDFofBaseDocument[term,
inverseDocumentFreq]
Declare a HashMap variable TFWeightOfBaseDoc[term, termFreqWeight]
Declare a HashMap variable TFWeightOfQueryDoc[term, termFreqWeight]

DocumentTermFrequencyMap= computeTermFrequency()
IDFofBaseDocument=
computeInverseDocumentFrequency(DocumentTermFrequencyMap[BaseDocu
mentID])
TFWeightOfBaseDoc=
computeTermFrequencyWeightOfDocument(DocumentTermFrequencyMap[Ba
seDocumentID])
        FOR number of QueryDocuments of ListOFQueryDocumentID
        TFWeightOfQueryDoc=
        computeTermFrequencyWeightOfDocument(DocumentTermFreque
        ncyMap[QueryDocumentID])

                FOR number of terms of TFWeightOfQueryDoc
                SUM SimilarityValue=TFWeightOfQueryDoc*
        TFWeightOfBaseDoc* IDFofBaseDocument
                END FOR
                STORE OkapiValue[QueryDocumentID,
        SimilarityValue]
        END FOR
SORT OkapiValue[QueryDocumentID, SimilarityValue] and select TopK
QueryDocumentID's
RETURN QueryDocumentID's
END computeOkapiSimilarity

computeTermFrequency()
Declare a HashMap variable DocumentTermFrequencyMap[Document ID,
Map of Terms of the Document]
        FOR     number of documents
        Declare a HashMap variable TermMap[Term, Frequency of the
Term in the document]
                FOR     number of terms in the document
                Calculate term frequency of the term in document
                STORE TermMap[term, term frequency ]
                END FOR
        STORE   DocumentTermFrequencyMap [Document ID,
TermMap]
        END FOR
RETURN DocumentTermFrequencyMap
END computeTermFrequency

computeInverseDocumentFrequency(TermMap[term, term frequency ])
Declare a HashMap variable InverseDocumentFrequencyMap[term,
inverseDocumentFreq]
        FOR     number of terms in TermMap
        Calculate inverse document frequency of the term
                IF inverse document frequency of the term < 0 THEN
                inverse document frequency of the term=0
                END IF
        STORE   InverseDocumentFrequencyMap[term,
inverseDocumentFreq]
        END FOR
RETURN InverseDocumentFrequencyMap
END computeInverseDocumentFrequency

computeTermFrequencyWeightOfDocument(TermMap[term, term frequency ],
DocumentLenght)
Declare a HashMap variable TermFrequencyWeightMap[term,
termFreqWeight]
        FOR     number of terms in TermMap
        Calculate term frequency weight (term frequency, DocumentLengt
        STORE   TermFrequencyWeightMap[term, termFreqWeight]
        END FOR
RETURN TermFrequencyWeightMap
END computeTermFrequencyWeightOfDocument
```

ACKNOWLEDGMENT

:nis ongoing research project is collaboration between the :niversity of Maryland and the Lanka Software Foundation id it is partially funded by the Robert H. Smith School of .usiness, University of Maryland and the United States tational Science Foundation under grant IIS1237476. The :thors would like to thank Louiqa Rashid for her guidance, id Joe Langsam (University of Maryland), Mike Bennett :DM Council) and Mark Flood (Office of Financial Research, :S Department of the Treasury) for feedback.

The authors also thank WSo2 Inc. for providing hosting :cilities for web applications and databases.

Software developed on the Karsha FOSS project is made ailable under AGPL license at the following site: :ps://github.com/Karsha-Project-LSF/Karsha-Annotate

REFERENCES

(2013) The Electronic Municipal Market Access system website. [Online]. Available: http://emma.msrb.org/

(2013) The Consumer Financial Protection Bureau website. [Online]. Available: http://www.consumerfinance.gov/

(2013) The U.S. Securities and Exchange Commission website. [Online]. Available: http://www.sec.gov/

A. Gingrande ,Processing Unstructured Documents: Challenges and Solutions. AIIM User Guide; AIIM International, 2004, ch. 2, pp.5-10

Guan-yu LI, Sui-ming YU and Sha-sha DAI, " Ontology based query system design and implementation", International conference on network and parallel computing, pp.1010 -1015, 2007

(2013) The Financial Industry Business Ontology website. [Online]. Available: http://www.hypercube.co.uk/edmcouncil/

Burdick, D., Hernández, M., Ho, H., Koutrika, G., Krishnamurthy, R., Popa, L., Stanoi, I., Vaithyanathan, S. and Das, S., "Extracting, Linking and Integrating Data from Public Sources: A Financial Case Study," IEEE Data Engineering Bulletin, Volume 34, Number 3, pages 60-67, 2011.

Mauricio A. Hern´andez, Howard Ho, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, Ioana R. Stanoi, Shivakumar Vaithyanathan, Sanjiv Das ,Unleashing the Power of Public Data for Financjal Risk Measurement, Regulation, and Governance, IBM Technical Report, 2012.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., McCurley, K. S., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J. Y.: A Case for Automated Large-Scale Semantic Annotation. Journal of Web Semantics, 1(1) 115–132, 2003.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Elsevier's Journal of Web Sematics, 2(1), 2005

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: Towards Semantic Web Information Extraction, in proc. International Semantic Web Conference (ISWC) 2003.

Mehrnoush Shamsfard, Azadeh Nematzadeh and Sarah Motiee, " ORank: An Ontology Based System for Ranking Documents", International Journal of Computer Science, vol .1, pp.225- 231, 2006.

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. Mach. Learn., 34(1-3).

S. Strassel, D. Adams, H. Goldberg, J. Herr, R. Keesing, D. Oblinger, H. Simpson, R. Schrag, and J. Wright, "The DARPA Machine Reading Program - Encouraging Linguistic and Reasoning Research with a Series of Reading Tasks", ;in Proc. Language Resources and Evaluation(LREC), 2010.

R. Wilkinson, J. Zobel, R. Sacks-Davis, Similarity measures for short queries, proceedings of the Fourth Text Retrieval Conference, pages 277-286, Gaithersburg, Maryland, 1995.

C. Jonquet, P. LePendu, S. M. Falconer, A. Coulet, N. F. Noy, M. A. Musen, and N. H. Shah. NCBO Resource Index: Ontology-Based Search and Mining of Biomedical Resources. In proc. Semantic Web Challenge, 2010.

D. Trieschnigg, W. Kraaij, and M. J. Schuemie. Concept based document retrieval for genomics literature. In proc. Text REtrieval Conference (TREC) , 2006.

Nadzeya Kiyavitskaya, Nicola Zeni, James R. Cordy, Luisa Mich and John Mylopoulos Semi-Automatic Semantic Annotations for Web Documents, Proc. of Semantic Web Applications and Perspectives, 2nd Italian Semantic Web Workshop (SWAP 2005), Trento, 14-15-16 December, 2005.

[19] G.Palma, M. Vidal, E. Haag, L. Raschid, A. Thor. Measuring Relatedness Between Scientific Entities in Annotation Datasets. Technical Report, University of Maryland, 2013

[20] G. Palma, M. Vidal, L. Raschid, A. Thor. Comparing the Disease Signatures of Drugs Using Shared Annotations and Ontological Relatedness, Technical Report, University of Maryland, 2013