Semantic Information Retrieval Architecture for Unstructured Data using NLS

Jongwook Woo
Associate Prof., Department of Computer Information Systems, California State University Los Angeles
jwoo5@calstatela.edu

ABSTRACT
This paper proposes a semantic search system with the NLS (Natural Language Semantics) query parser and metadata tagging for unstructured data, which establishes high quality search, that is, information retrieval. Thus, the proposed system collects data from unstructured data sources and adds tags to the data as metadata. And, data with metadata becomes structured data with semantic information, which makes the data analysis much easier. And, the system parses the NLS input by the user to retrieve information from the search engine. Then, information result with search engine follows the common format that is easy to read and analyze. Besides, it will be clearly illustrated how much risk of the systems has been mitigated relative to the production of a full prototype systems.

Keywords: Unstructured data, Solr, UIMA, NLS

1. INTRODUCTION
Current data set across the world are large volumes of data in the format of semi-/un-structured or structured data. Structured data is relatively easy to extract information as it contains the semantic information of data. However, unstructured data is not easy to extract knowledge as it does not contain any structured semantics. Therefore, military intelligence and operations staffs spend too much time to prepare reports and briefings by searching for related information from many data sources, to estimate data relevance, and produce information products by fusing and formatting relevant data.

Thus, there are needs to extract semantics information from unstructured data, which will improve the understanding of the broader “tell me about” query of the user and the way to index and retrieve the relative data efficiently, especially in the collection groups of data in persons, events, groups, places etc. at the area of defense and homeland security. Furthermore, the systems output should look like a finished information product so that the users can understand the knowledge more easily. This proposal illustrates the research plan about a system that builds semantic search applications from un-/semi-structured data through metadata tagging capabilities and it also understands the user’s semantic and natural language query to extract knowledge from the search engine. And, the search result should be the packaged as a finished information product.

Section 2 describes the related work done for unstructured data retrieval. Section 3 summaries the proposed architecture. In Section 4, we describe the tasks to build the systems. We conclude the research in Section 5.

2. RELATED WORK
Peñas et al [3] illustrates that text omit an important information and it can be supported by background knowledge. Thus, they synchronize NLS and a knowledge representation (KR) scheme. Golub et al [4] proceed EnTag project that is to investigate the effect on index and retrieval when using only social tag and using both social tag and proposed vocabulary. Hakrae et al [5] presents an algorithm to reuse the existing tags that are related. Pyu et al [6] has an algorithm that efficiently extracts contents from untagged texts using Link Grammar Parser and filter the final extraction rules using Wordnet and linguistic patterns.

3. PROPOSED ARCHITECTURE
Before Internet and Web, we did not have enough data for analyzing people, society, and science etc. Contradicting to the past, after Internet and web, it has become more difficult to analyze data because the data is in the multi-modal data types that are un-/semi-structured text, image, video, and audio etc. Therefore, the areas of Defense and Homeland Security have the difficult task of quickly extracting knowledge, understanding, and determining the crucial information from complex data.

There are mainly three approaches to extract information from data. The first is information retrieval on search engine or RDBMS (Relational Database Management Systems) with query by the user. Traditionally, RDBMS stores structured data and some data fields are indexed to be searchable. However, it overloads RDBMS and causes the performance issue, especially, in text search so that dedicated search engine has received highlights and used for information retrieval in many institutes, which is to retrieve data from the search repository in the order of high relevancy. It can be either web search to retrieve data from the web like google.com and bing.com etc or local search to retrieve data from the local database using search engine such as FAST, Solr, and ThinkingSphinx, etc.

The second is text classification that classifies data or documents into specific classes with training data by executing one of algorithms: Naïve Bayes, Vector Space, and Support Vector Machine (SVM) etc. The third is to extract knowledge by annotating documents with metadata, which is the semantic information of the unstructured data. It needs training data with ontology in the specific domain to extract
information to create metadata, which is developed with UIMA and Amilcare, etc.

Many open sources and freewares are illustrated in this section since they are adopted in the proposed systems as they are proven in the market and many vendors have used them commercially. There have been many discussions about the security and safety of the open sources in the community. And, most people believe that the open sources are much safer than the proprietary codes because many developers are monitoring the open codes together. In other hands, we never know what will be embedded in the proprietary codes by the developers’ intentions.

In the paper, we integrate the first and the third approaches above and illustrate the possible plan to convert the unstructured data to structured data with metadata by analyzing semantic information with ontology in a domain. Besides, we investigate and develop the NLS query parser and search result format in order to parse the user’s semantic query and produce the common output format respectively [6, 7, 12, 13]. Then, we also describes the work plan how possibly to retrieve relevant data in the format by the parsed query from the search repository of the data with metadata.

Figure 1 is the conceptual overview of our proposed system that is composed of four functions: Data Sources, Unstructured Information Management Framework (UIMF), Search Engine with data and metadata, and Input/Output with NLS Query Parser and Search Result Format Builder. The tasks to implement the functions are described in the following sections.

Many open sources and freewares are illustrated in this section since they are adopted in the proposed systems as they are proven in the market and many vendors have used them commercially. There have been many discussions about the security and safety of the open sources in the community. And, most people believe that the open sources are much safer than the proprietary codes because many developers are monitoring the open codes together. In other hands, we never know what will be embedded in the proprietary codes by the developers’ intentions.

3.1 Unstructured Data collection
It is one of issues to identify relevant multi-modal data sets that incorporate various forms of text and visual data. Fortunately, there are many public data sets that can be used as input data for the system. We will collect unstructured data from the public data set provided by Google, Amazon, Datamob, and US government etc [15-19]. Most of public data set is text data but there are multimedia data provided by US government [19]. Public dataset is collected in the format of XML and JSON, which is structured data that is collected by given data feed APIs. However, the research is mainly to extract information from unstructured data so that we will look up and collect unstructured data set from the public data set in a community/ domain.

Besides, many internet companies provide proprietary APIs that enable the third party developers to use and integrate the APIs to provide the proprietaries’ data to the users. One of famous examples is Google Map API that provides Google’s geo data to the users. And, many other companies follow the Google so that yelp.com and citysearch.com provide APIs that enable their search function to provide search result data to the users [30-32].

3.2 Metadata Extraction from Unstructured Data
There are number of approaches to analyze and extract information from unstructured data such as Apache UIMA [10], Google ClearTK [11], and Apache Nutch [12] etc.
Especially, UIMA supports functions to extract metadata from a document and use them as annotations, that is, tags. Besides, UIMA provides a manual annotation to determine the domain and to define types of metadata. By studying and adopting these pre-existing systems, we can investigate the related technologies and develop an optimal method to run the application with supporting tags.

Metadata is to summarize the information of data from the data sources. We design a data structure to store data as structured data with metadata in XML, JSON, or DB format converted from the multiple data sources by UIMF of Figure 1, which is described below. Since data have a relevant topic in a domain, we need to create an ontology that best reflects the community by understanding their shared data.

Ontology includes data categorization schemes, thesauruses, vocabularies, keyword lists, and taxonomies. An ontology is a specification of the object of interest in a domain, which describes an object, its properties, and relations between objects. A taxonomy is a classification selected by “is-a” relation of the ontology. A vocabulary as a metadata is a specification that describes what language is used in the ontology, the ontology itself, and its taxonomy. We only need metadata to analyze data but, if needed, relations between objects will be considered to be annotated.

```
<types>
  <name> Article </name>
  <features>
    <name> title </name>
    <name> director </name>
    <name> actors </name>
    <name> description </name>
    <name> publish_date </name>
  </features>
</types>
```

Figure 2. Example Metadata for Movie Article

In recent years, domain experts have developed ontologies on behalf of the artificial intelligence laboratories. Many domain experts develop standardized ontologies that they can share and annotate in their fields. Thus, we can design new ontology for a domain but we can also use the existing ontology if it is the good match for the domain interests in order to save resources and utilize its safety. Once the ontology is developed or adopted from the existing ones, it can be stored at the metadata registry and used to build structured data. Data from the heterogeneous data sources need to be converted to the data structure of the ontology and stored at the DB or file systems in XML or JSON etc.

As shown in Figure 1, UIMF is to convert unstructured multi-modal data to structured data by extracting metadata and tag from the unstructured data of a domain. For example, if there is an article about a movie in movie domain, UIMF enriches the article with metadata contained in the text such as title, director, and actors etc as ontology in movie domain. UIMF provides Annotation Function (AF) that is to label a region of text. For instance, in the text “Tom Cruise has been nominated for three academy awards”, “Tom Cruise” in characters 1 to 10 is annotated as “Person” by AF. Besides, we need to define Type Systems (TS) in Domain Model (DM), which is manually specified by domain experts. Type should be easily created by GUI so that we need to implement a user friendly GUI. In TS that is composed of Types, we can create an Article Type to extract metadata from articles with annotations (title, director, and actors etc). Figure 2 illustrates a possible structure of metadata for movie articles.

3.3 Data Search with Search Engine

Data with metadata from UIMF is the structured data that is extracted from the unstructured input data. For the small number of data – less than 1 million entities, we can use RDBMS’s index function and query to retrieve the data. However, we are talking about efficient data and text search, which can be efficiently indexed and searchable by existing open source search engines.

We implement the search functions by adopting the existing popular search API Lucene [13] and Solr [14] server. Solr has been used many companies such as AOL, citySearch.com etc as a search server implemented on Lucene API. Thus, metadata can be indexed and uploaded to the repository of the search engine.

Therefore, users can retrieve the data with more accuracy as the metadata contains data to search. Once the elements of the metadata are indexed by search engine, users can search and retrieve the data from the search engine’s repository.

Figure 3 is the possible proposed architecture, especially of the search function, which illustrates the search function in detail as briefly shown in Figure 1. It is composed of Solr search server and Service Engine. Solr server stores structured data with metadata of UIMF and indexes metadata including tags. SearchService is to provide a search GUI using search API given by Solr. Solr server posts the XML files to Solr repository so that SearchService can search and retrieve data of the repository.

Service Engine is composed of many services that will be implemented: IndexService, PostService, SearchService, SecurityService, UIService, and DataAccessService. IndexService is to index searchable data from structured data with DataAccessService and to generate XML files. PostService is to post the XML files to the Solr repository. SearchService takes tags and category inputs from the user and searches data at the repository and returns the data retrieved to the user. SecurityService is to register users with user names and passwords for authorization and
authentication. DataAccessService provides connection to the structured data with metadata. UIService presents the User Interface (UI) pages and its related functions to the users for the search.

3.4 Search Input/Output Data
Semantic input, for instance, “tell me about …”, from the user can be processed with Natural Language Semantics (NLS) query parser as a query input to the search engine, which we can extend from the existing APIs. For example, Solr provides Query Parser API so that we will investigate and develop our NLS and query parser to add it to the Solr query parser [12, 13]. Using the parsed input, result data can be retrieved from the search engine.

Most search engine generates search result in the format of XML, JSON, and HTML. However, in the topic of the proposal, output data of the search result is required to be in the format of the commonly used, e.g., Microsoft Word document. There are many ways to convert HTML document to word DOC. For example, we can use Microsoft.Office.Interop.Word API and save it in *.doc file type. And, we can also use a combination of openXML 2.0 and AltChunks to insert the formatted HTML to generate word doc document [6, 7].

4.IMPLEMENTATION TASKS
Figure 1 is the conceptual overview of our proposed system that is composed of three functions: data sources, Unstructured Information Management Framework (IMF), Search Engine with data and metadata, and Input/Output as semantics query parser and search result format builder. The tasks to implement the functions are described in the following sections.

The proposed work consists of the following five tasks, which are detailed in the following sections:
• Task 1 – Collect and Evaluate Unstructured Data
• Task 2 – Design Unstructured Information Management Framework (UIMF)
• Task 3 – Design Data Search Systems
• Task 4 – Construct semantic query parser and search result format builder.

4.1 Task 1: Data collection
Task 1 is to collect data from n multiple data sources DS1, DS2, …, DSn at the different nodes or web sites, which contains text, html, video, audio, and image data for tactically accurate decision.

The multi-modal data can be either structured or semi-/unstructured data but we mostly consider the unstructured data as input in the same domain because the structured data in the domain is easy to extract metadata that includes tag data. As
discussed at the background section 2.1, we will collect multi-modal data from the public data set provided by Google, Amazon, Datamob, and US government etc. We will look up and collect unstructured public data set in the domain out of the public data set [15-19].

4.2 Task 2: UIMF (Unstructured Information Management Framework)

Once data set is collected, the task 2 is to determine an effective tagging technique by designing ontology for metadata (and tag) in a domain and that can be used to convert unstructured data to structured and to increase accuracy of information systems.

As shown in Figure 1, UIMF is to convert unstructured multi-modal data to structured data by extracting metadata and tag from the unstructured data in a domain. The example is show in the background section 2.2.

4.3 Task 3: Design Data Search Systems

In task 3, we design search engines by adopting the existing search systems and demonstrate the ability to search and/or navigate the tagged data with a search engine. Data with metadata from UIMF is the structured data that is extracted from the unstructured input data. Thus, it can be uploaded to the repository of the search engine and indexed. Therefore, users can retrieve the data in high accuracy with the indexed metadata that contains data to search.

Once the elements of the metadata are indexed by search engine, users can search and retrieve the data information from the search engine’s repository. We can implement the search functions with the existing popular search engine, Lucene [13], and Solr [14] server. Solr has been used many companies such as AOL, cisySearch.com etc as a search server implemented on Lucene engine.

And, once the search function is implemented, we have to tune and experiment the system by monitoring how relevant and accurate the search results. That is, we need to investigate how well metadata is built and then need to tweak the system to achieve the better relevancy.

4.4 Task 4: Input and Output

Task 4 is to investigate and develop NLS query parser to parse the semantic query by the user as input system and to implement the search result format builder in order to display the search result more user friendly in the format of Microsoft word as output.

As discussed in the section 2.4, we will extend the existing tools such as Solr query parser API to analyze the input semantics and Microsoft API with OpenXML 2.0 and AltChunk etc in order to convert search results to Microsoft Doc format.

5. CONCLUSIONS

The paper presents system architecture that extracts and retrieves data from unstructured data set. The system is composed of Solr search engine, NLS, tagging systems. Thus, the collected unstructured data is converted to structured data by tagging. Then, when the user enters input as a normal conversation, NLS parses it and retrieve tagged data using the Solr search engine.

REFERENCES


“OpenLink AJAX Toolkit”, http://oat.openlinksw.com/

“Improving Intranet Search-Engines Using Context Information from Databases”, Christoph Mangold, Holger Schwarz, Bernhard Mitschang, Proceedings of the 14th ACM international conference on Information and knowledge management, 2005

“e-Business Architecture with CMS and Search Engine”, Jongwook Woo, The 2009 19th Annual KSEA South-Western Regional technology Conference (KSEA-SW 2009), Irvine, CA (Feb 7, 2009)


CityGrid APIs, http://docs.citygridmedia.com/display/citygridv2/CityGrid+APIs

Yelp APIs, http://www.yelp.com/developers/documentation

Meetup APIs, http://www.meetup.com/meetup_api/

AUTHOR PROFILE

Jongwook Woo is currently an Associate Professor at Computer Information Systems Department of California State University, Los Angeles. He received the BS and the MS degree, both in Electronic Engineering from Yonsei University in 1989 and 1991, respectively. He obtained his second MS degree in Computer Science and received the PhD degree in Computer Engineering, both from University of Southern California in 1998 and 2001, respectively. His research interests are Information Retrieval/Integration/Sharing on Big Data, Map/Reduce algorithm on Hadoop Parallel/Distributed/Cloud Computing, and n-Tier Architecture application in e-Business, smartphone, social networking and bioinformatics applications. He received teaching/research grants from Amazon AWS and owns Academic Partnership with Cloudera in Big Data. He has published more than 40 peer reviewed conference and journal papers. He also has consulted many entertainment companies in Hollywood.