Pull and Push Business Functions in an Experimental Text Processing System

Bogdan D. Czejdo, Center for AMEDD Strategic Studies, Fort Sam Houston, TX 78234, USA; E-mail: bogdan.czejdo@amedd.army.mil
L. Harrison Hassell, Center for AMEDD Strategic Studies, Fort Sam Houston, TX 78234, USA
Barbara Wojcik, Center for AMEDD Strategic Studies, Fort Sam Houston, TX 78234, USA;

ABSTRACT

We observe an unprecedented growth in the volume of unstructured data. Active use of business information contained in large volumes of unstructured data is becoming one of the biggest challenges now. In this paper, we examine an automated mechanism for pull and push functions for business documents. The pull function allows users to access the information contained in documents. The push function alerts users to the presence of information contained in new documents which is consistent or inconsistent with the background information. To implement these functions, documents are annotated using XML tags, and then, XML query processing techniques are used. Our approach is to use a limited context defined by an ontology or set of well established background documents for the guidance in identification and annotation of basic concepts and relationships in new business documents. The described text mining system is a highly modular and parametric, giving the human a tool to adjust quickly in a dynamically changing environment. The indirect goal of this paper is to provide a foundation for a new self-tunable text mining system that can adjust to new environment by itself.

1. INTRODUCTION

The abundance of business documents makes it increasingly likely that the precise information the user needs or wants is available. At the same time, however, retrieval of this information is much more challenging. Fortunately, this trend has been accompanied by unprecedented progress in technologies for content-based access to text documents. Using these technologies is crucial for achieving competitive advantage for our businesses.

Current information extraction techniques are either keyword/category based, such as Google, AltaVista [1] or Yahoo [8, 13], or structure dependent such as Rapper [10] and XWrap [9]. Our approach to text mining is to use the combination of qualitative and quantitative techniques for identification of relevant concepts and relationships. The selected concepts and relationships are extracted in the process guided by ontologies and by background documents for the domain of interest.

In this paper we address the problem of simultaneous text mining of two distinct groups of documents, background documents and new documents, as shown in Fig.1. The documents will be also referred as text corpora. We examine an automated mechanism for pull and push functions for business documents. The pull function allows users to access the information contained in the documents. The push function alerts users to the presence of information contained in new documents which is consistent or inconsistent with the background information. To implement these functions, documents are annotated using XML tags, and then, XML query processing techniques are used.

The described system is highly modular and parametric giving the human a tool to adjust quickly to text mining in a dynamically changing environment. The indirect goal of this paper is to provide a foundation for a new self-tunable text mining system that can adjust to new environment by itself.

Our approach assumes multi-stage document processing. These stages include ontology processing, background document annotations, queries for ontology and background documents, new document annotations, comparative queries for new and background documents, and generation of alerts when the information in new documents does not match information the background documents.

XML was chosen as a language for our annotations [4, 12] for several reasons: (1) XML provides a simple, standard, self-describing way of storing and exchanging text and data, (2) XML-based retrieval systems are relatively simple and can retrieve XML information quickly without linguistic analysis of text documents during the query time, (3) XML notation is very convenient as an internal representation because it allows for incremental annotations that represent explicitly the various phases of information extraction (knowledge discovery process), (4) XML can provide many annotation types and XML-based query systems can retrieve only those types of information that are requested, (5) XML annotations can be inserted/deleted/modified in the future responding to dynamically changing needs, (6) XML annotated text is open to further processing beyond annotations, and (7) XML texts can be integrated with other structured data.

The results of our project represent an integration of various text processing technologies and have immediate application for pull and push functions as shown in Fig.1. These results generalize beyond our application and will be important wherever concept-based information retrieval through XML-capable search engines or query systems is desirable.

The "Pull" functional requirement is the ability of the system to answer queries for new or background documents. The "Push" functional requirement needs more discussion. In the business world we often have to deal with new events described in new documents that need to be reflected directly in the "knowledge base" of our corporation. The text processing system can help us identify such documents to be included in the set of background documents. We refer to such function of text processing system as a push function.

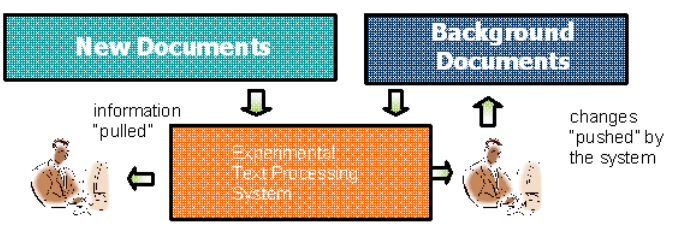
The system described in this report is being designed to support military medical documents; however, such a system, because of its general nature, can be successfully used in other business environments. It provides integrated concept-based access and awareness of unstructured data. The system is capable of querying unstructured data sources and continuous monitoring of event patterns in new documents.

2. FLEXIBLE ARCHITECTURE OF EXPERIMENTAL TEXT PROCESSING SYSTEM

Our experimental text processing system is flexible in the sense that it can be constructed from various available modules. Each module is highly parametric allowing the human to adjust parameters for the existing needs.

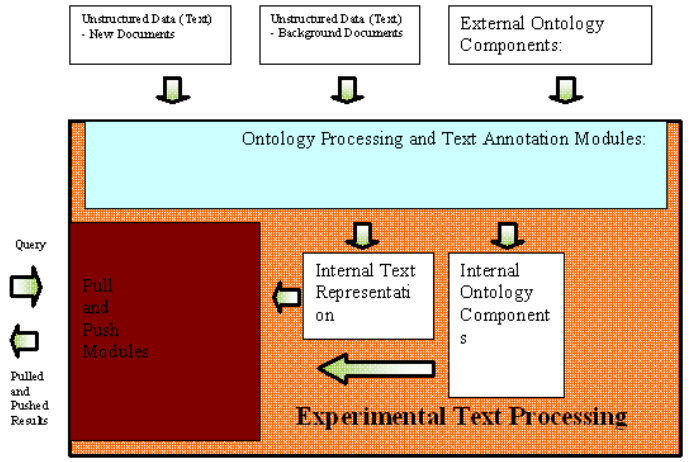
The architecture of our experimental text processing system is shown in Fig 2. It is constructed from relatively independent modules: Ontology Processing Module, Text Annotation Module, and Pull and Push Modules.

Figure 1. An overview of pull and push functions



Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 2. An overview of a flexible system architecture of an experimental text processing system



Text annotation can involve concepts hierarchy of concepts and concepts relationships. For each annotation level we can apply various statistical processing types, for example frequencies of words, direct collocations, and distant collocations. For each pair of annotation level and processing type, we can use various ontology components: simple and compound concepts, relationships, word stems, synonyms, antonyms, stop words, stop modifiers, etc. Not all combinations of system parameter values are important to consider. Some combinations of parameter values are useful in specific situations only, and some combinations of parameter values can result in either contradicting functionality of the modules.

Based on internal representation, the query can be issued or alerts generated. The query can be:(a) keyword search, (b) hierarchy and relationship search, or (c) comparative.

The internal text annotation is done using XML. XML allows us to annotate important concepts, hierarchy of concepts and relationships between them. These annotations are done using tags ***<concept>*** and ***<relationship>*** and various tag properties e.g. ***type, source,*** etc.

Document processing is done in several stages. In the first stage, the internal ontology is created by using components of external ontology (if available) and statistical processing of primary background but also new documents as shown in Fig. 3. In the second stage, relevant concepts, hierarchy of concepts and relationships between concepts are identified in background documents and annotated using XML as shown in Fig.4. In the third stage, the query is processed to verify the consistency of knowledge contained in ontology and background documents. In the fourth stage, the relevant concepts, hierarchy of concepts and relationships between concepts are identified in new documents and annotated using XML as shown in Fig.5. In the fifth stage, the alerts are generated if the consistency of knowledge contained in new documents and background documents is violated.

The flexibility of the architecture of the text processing system is of crucial importance for success in business automatic information extraction from text. So

Figure 3. Stage 1of text processing

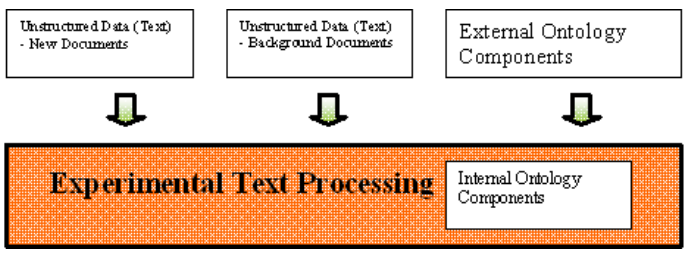


Figure 4. Stages 2 and 3 of text processing

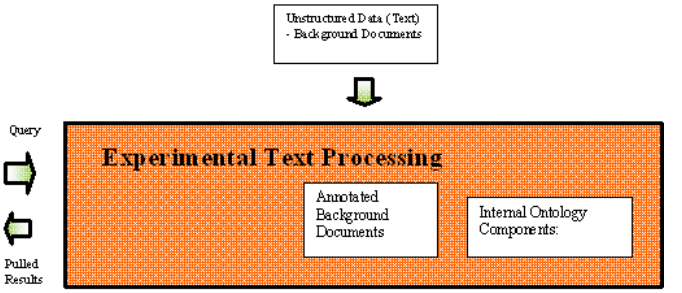
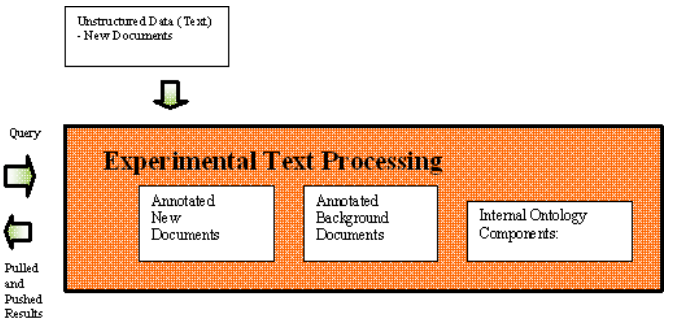


Figure 5. Stages 4 and 5 of text processing



far there are no black box solutions that are good for text processing for a broad range of styles of business documents. Our experience suggests that in order to succeed in a complex business environment with a text processing system it is necessary to use an iterative approach in terms of both domain and complexity of analysis to tailor the system to business needs. We suggest proceeding concurrently with experiments in these two directions.

3. ONTOLOGY PROCESSING

Whenever we humans process an individual document (text) we view it in a broad context of facts and rules acquired throughout our life. Let us refer to this context as a background knowledge. The background knowledge can be in various forms. It can be in the form of background documents that we remember or studied and can recall quickly. In can be also in the form of more digested information such as dictionaries. Let us refer to "digested" knowledge as an ontology. In general, the ontology can contain not only the list of concepts as in simple dictionary but also explicit concept hierarchy and relationships between concepts.

Our understanding of a new document very much depends on the "scope" of our background knowledge. Obviously, there are various scopes of background knowledge and "quality" of background knowledge can be different for different areas. Let us, in this section, concentrate on a "digested" knowledge – ontology. When we refer to an ontology, we can mean a general ontology or more often, an ontology for the specific area of interest e.g. medical, technical.

One of the measures of ontology "quality" is ontology completeness levels. To compute a concept completeness level for an ontology with respect to a set of documents, we first compute the number of all concepts in ontology that are found in a set of documents and divide it by a number of all concepts in a set of documents. Similarly, we can define concept hierarchy completeness level and relationship completeness level. By completeness level, we can mean overall completeness level or completeness level with respect to a subset of documents or even a completeness level with respect to single document or its fragment. This is an important measure how "good" ontology is for the specific set of documents. When several distinct sets are processed e.g. new and background documents the completeness measure for each subset may need to be calculated.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

3.1 Concepts in Ontology

There are different requirement for ontologies depending on text recognition levels. For the lexical text recognition level, the ontology should contain the list of concepts, their possible representations and the methods to identify them. Specifically, an ontology for lexical recognition can contain: (a) list of concepts represented by primary names, alternative names, stem words, and synonyms, (b) list of stop words, and if applicable (c) list of misspellings. For the list of concepts, we assume that each concept is represented by a unique primary name, e.g. "**organization",** that we call *name*. Each concept *name* is associated with a list of alternative names, e.g. "**organizations,"** to allow for alternative forms, singular and plural, etc. The list of alternative words can be also generated from the stem words, e.g. assuming a stem word "**organization,"** the alternative name "**organizations"** can be generated by appending the "s" to the stem word. In general, both generated and stored alternative words can be used. Each concept has a list of synonyms, e.g. "**corporation"** for the concept "**organization."** Each concept can have a list of misspelled words, e.g. "**corporation."** The list of misspelling words can be also generated from the alternative names by using some rule e.g. removing one character, replacing one letter by another letter. In general, both generated and stored misspelled words can be used.

3.2 Creating Ontology with Concepts

Internal ontology can be built from a variety of external resources. In the medical area, there are many existing external ontologies e.g. Mesh, UMLS, SNOWMED. Some of the external ontologies are very large e.g. UMLS, so that they need to be appropriately processed (restricted in size) to be useful for text recognition purposes. Sometimes, for a narrow area of interest, the internal ontology can be created from the specific set of documents (text corpora). This process is called ontology extraction from text corpora. For our project text corpora can contain both background documents and new documents.

Ontology extraction is based on finding statistically important words in documents. The words in the document are also referred as tokens. It is easy to see that ontology extraction, guided only by global statistical computations, may not be reliable, especially when we deal with a small number of documents or groups of documents with significantly different presentation features. Therefore, statistical techniques should take into consideration a diversity of documents. In summary, when an ontology is not available, there is a potential increase in the number of generated errors, in terms of false positives, but human interaction can make this process much more reliable.

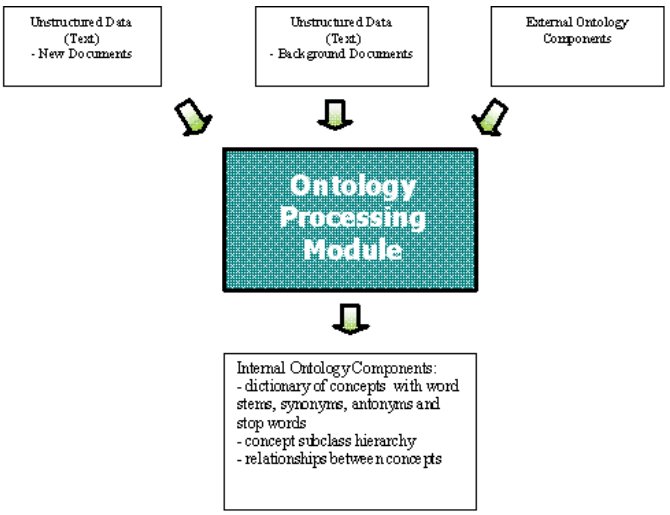
In general, the external ontology components can be used together with internally created ontology for lexical text recognition. More specifically, the optimal use of ontology might require appropriate combining of external and internal components so that they will provide both effective and efficient support for the text recognition.

Let us discuss briefly the process of creating the internal ontology from the documents (text corpora) as shown in Fig. 6. As we discussed before, there is need to construct such internal ontology when the external ontology does not exist for a given area, or when external ontology needs to be enhanced by knowledge from the given area (an external ontology has a low completeness level with respect to text corpora). The process of constructing internal ontology is multi-phased. First, it involves finding the frequency of all tokens in text corpora. Second, we eliminate some tokens of no semantic value e.g. "a", "the", and create a stop list from them. Third, we choose mainly nouns that can be potentially relevant concepts. Next, we need to group tokens representing the same concept together and assign to the group a unique identifier i.e. typically *primary name*. Last, we choose some frequency threshold to eliminate irrelevant concepts. An enhanced version of this multi-phase process will also involve computing frequencies for collocated tokens.

Let us also discuss some specifics about combining of external and internal components so that they will provide both effective and efficient support of text recognition. We can enhance the multi-phase process of creating internal ontology, described above, by investigating frequency of co-occurrence of external ontology concepts with text corpora concepts to identify new concepts with multi-component names.

In summary, the ontology processing module creates internal ontology components from modified external ontologies and from components obtained by ontology extraction process from text corpora. Actually, this process is much more complicated than described above. Let us mention the aspect of ontology

Figure 6. Ontology processing



self-modification. After the internal ontology is used for text recognition, it should self-adjust automatically e.g. re-arrange, expand and possibly shrink. This self-adjusting mechanism, together with zooming techniques, will allow ontology to respond to dynamically changing text environment.

At this stage let us assume that the example internal ontology is available and it contains the following concepts: "**organization,"** "**disease**," "**rate**," and "**study**." The concept "**organizations,**" has an alternative name "**organizations**" and the synonym "**corporation."** Similarly, alternative name for **"rate"** is **"rates,"** and for **"injury"** is **"injuries"** etc**.**

3.3 Multi-Component, Compound, and Derived Concepts

So far we were discussing simple concepts whose primary name or its equivalent textual representations contain one word. There are also concepts whose primary name or one of its equivalent textual representations contain several words (components). We will refer to such concepts as *multi-component* concepts. If at least one of the components corresponds to another concept, then the *multi-component* concept becomes *compound* concept. If all components of *multi-component* concept match some other ontology concepts we will call it a *derived* concept.

Let us consider a sample ontology from Section 3.2. If is enhanced by a new concept, "injury rate," that concept would be classified as *compound* since "rate" is already a valid concept. If we add another new concept "injury," then the *compound* concept "injury rate" would become also *derived*. It is important to notice that as ontology changes the properties of concepts can also change. In a well designed system all changes need to propagate appropriately.

The *multi-component* concepts can be stored and processed in the ontology as simple concepts. The *compound* and *derived* concepts can be also stored as *simple* concepts but then some additional ontology processing is necessary to discover relationships between then. Alternative notations are described in the next subsections.

3.4 Hierarchy of Concepts

Multi-component concepts are candidates for building traditional type hierarchy for concepts. For example, "injury rate" is a subtype of the concept, "rate." This classification can be explicitly specified in the ontology or can be computed.

In general, there are also other candidates for building traditional type hierarchy e.g. "flu" is a subtype of "disease." Such a hierarchy can not be simply discovered by collocation of words but need to be extracted using more sophisticated analysis.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

3.5 Relationships Between Concepts

An ontology can also contain relationships between concepts. The relationship can be unique for some concepts but very often they are classified into some classes called relationship types. For example, let us consider two important types of relationships: "***is-a,***" and "***has.***" The first relationship type represents an alternative notation for a hierarchy of concepts as described in the previous section. In our example instead of concept hierarchy can be described by the following relationships: "injury rate ***is_a*** rate" and "flu ***is_a*** disease." The second relationship type "***has***" is used to describe a strong association between concepts including ownership, main activity or components. Let us assume that for our ontology example, we have a relationship "CASS ***has*** study." From the existing relationships new concepts and relationships can be derived. They will be also called derived concepts or derived relationships. From the "***has***" relationship, we can create a derived concept such as "CASS study". There are also derived relationships obtained by so called inheritance operation.

4. TEXT ANNOTATION

Typical documents include a wide variety of concepts from physical objects to events and states of affairs. The text recognition process will extract this information. The first step in identifying and annotating all important information in documents is to recognize basic concepts. This process can be extended to identify and annotate the concept hierarchy and relationships between concepts as shown in Fig. 7. This process establishes the mapping between words in the text, also referred as tokens, and concepts (i.e. ontology concepts), concept hierarchies, and relationships. Once a word is mapped to the ontology concept, it is called a text corpora concept or simply text concept. The same applies to concept hierarchy and relationships between concepts. There are many ways to describe this mapping. One very convenient, technique is to annotate such words inside the text e.g. using appropriate XML tags. This annotation technique is used in our paper.

4.1 Annotation of Concepts Using Ontology

The concept text annotation sub-module is based on lexical analysis of text, i.e. it scans the text, extracts each word, and annotates the words matching the concepts in ontology with appropriate tags. The annotated words become text concepts. In the initial phase, a set of text concepts is simply a subset of ontology concepts.

Let us consider an example of a sample fragment of a document:

Our CASS organization is involved in a variety of studies. Disease rates are CASS main study.

The system can process the text with concepts properly annotated shown below. The lines without tags are not annotated. The characters in bold constitute the original text. The annotated text fragment is shown in Fig. 8.

Figure 7. Text annotation

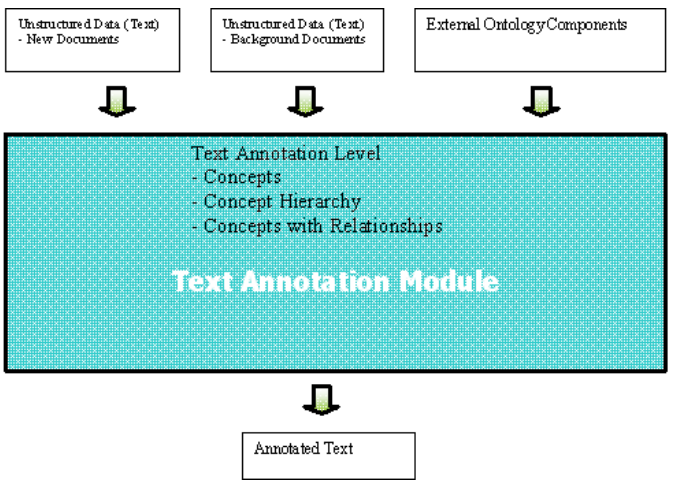


Figure 8. Document fragment with concepts annotated

```
1.  Our CASS
2.   <concept name="organization" >Organization </concept>
3.  is involved in a variety of
4.  <concept name="study"> studies </concept>
5.  .
6.  <concept name=" disease" > Disease </concept>
7.  <concept name=" rate" > rates </concept>
8.  are CASS main
9.  < concept name="study""> studies </concept>
10. .
```

Let us discuss annotations in Fig. 4 resulting from lexical text recognition process. In this process all words (tokens) matching an ontology concept are annotated as "concept" as shown in line 2, 4, 6, 7, 9 and 11. We assume that ontology contains all information necessary for an extended matching sub-process. Such a sub-process needs to take into consideration synonyms, misspellings, and alternative forms (e.g. singular and plural) resulting in a unique assignment of a word (token) to the concept. In actual implementation, the concept primary name can be a word stem but for the readability purposes we will use a meaningful name.

4.2 Annotation of Compound Concepts Using Statistical Computations

In the case the when ontology completeness level is low the process described in Section 4.1 can be augmented by finding words (tokens) with high frequency and treating them as concepts (text concepts) even though they are not present in ontology. We want include these concepts especially if they are collocated with the discovered ontology concepts.

The text annotation process for compound concepts can be performed together or after annotation of simple concepts in text. In the currently described process, all non-matching words are considered. First, we treat all existing ontology concepts in the text as a "hook." A word in the text with statistically supported collocations with the ontology concept is grouped together with the ontology concept itself to establish a new compound concept as shown in line 2-3 of Fig. 9. Such compound concepts are annotated as "compound". It is important to note that statistically supported collocations can return false positives in the sense that some unneeded components are attached to the beginning or to the end of the concept from the ontology.

In this process we want to make a more precise determination about concepts including compound concepts. We explicitly specify each concept property with the values: "simple," "compound," or "derived." Also, we can annotate the concepts by stating their source: they can be located in "ontology," or only in the "text" as shown in Fig. 9.

In our example, the concepts in text are annotated as the "simple" concept as shown in line 3 and 4 and as the "compound" concept as shown in line 2. The source of concept can be "ontology" as shown in line 2 and 4, and "text" as shown in line 3.

4.3 Annotation of Concept Hierarchy Using Ontology

In the case when ontology contains concept hierarchy, that information can be used in the annotation process of a document. Let us assume that "CASS organization" is recorded in the ontology as a sub-concept of "organization." Then the process of annotation can include this information as shown in Fig. 10.

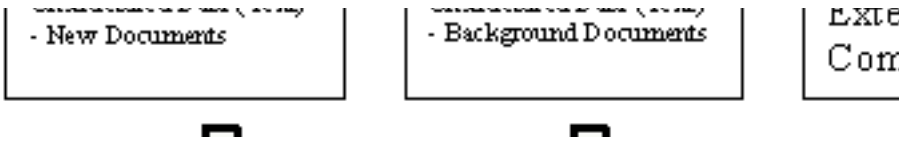
In our example, the concept "organization" has the property *hierarchy* with the value "super-concept" indicating that "CASS organization" is recorded in ontology as a sub-concept of "organization."

Figure 9. Annotations of compound concepts

```
1.  Our
2.  <concept name="CASS organization" type=" compound "  source= " text ">
3.          < concept name="CASS"  type="simple"  source=" text"> > CASS </concept>
4.          < concept name=" organization"  type="simple" source= " ontology " >organization </concept></concept>
```

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 10. Annotations of concept hierarchy



4.4 Annotation of Concept Hierarchy Using Statistical Computation

If an ontology with concept hierarchy is available, then concept hierarchies in documents can be identified and matched with ontology concept hierarchy. The problem is that many hierarchies in the document can be very complicated and they are only partially reflected in the ontology. There are different methods to deal with this problem. One method is to use syntactical text recognition by applying Parts of Speech (POS) analysis. If the document is written using strictly grammatical use POS analysis can be very helpful. In the document, however, that is not well structured, there may be an excessive number of modifiers since the parser could mistakenly cluster some other words together. In any case, POS can give us only some guidance but not precise results. Therefore, statistical computation (quantitative analysis) is required to obtain the relevant hierarchies. Typically, such annotation process is based on elimination rather than expansion. Statistically supported collocations are investigated as they serve the purpose of eliminating some irrelevant noun modifiers. In our case, for example, the modifier "our" is eliminated but the modifier "main" is included in the concept hierarchy as shown in Fig. 11.

4.5 Annotation of Relationships Based on Ontology

If an ontology with concept relationships is available, then the relationships can be identified and matched with ontology relationships. This process can be improved if a phrase structure parser is also available. Structure rules can refer directly to concept annotations. Let us discuss one of the rules: if the verb is 'is,' 'are,' etc., then name is 'is_a'; if the verb is 'announces,' 'has introduced,' etc. then name = 'has'; if it is 'includes,' exhibits,', etc., then type = 'capability,' and so on. In our example the verb 'are' is classified as 'is_a' type.

A relationship in the document can relate to the ontology in numerous ways giving several patterns for relationship identification. These patterns correspond to different sets of matching rules. There are many of these patterns. Let us discuss some of them. The first pattern is when the exact relationship type and concepts already exist in the ontology. For our example, the first pattern would be applicable if the relationship "disease rate ***is_a*** main study" exists in ontology. Another pattern would identify relationship type in the document if only one concept of relationship type can be matched.

4.6 Annotation of Relationships Based on Statistical Computations

There are many relationships used in different contexts and, therefore, the strict ontology structure may not be sufficient to retrieve the relevant relationships. The statistical matching rules can be used in such situations. If the statistical rule indicates a close association of concepts, then the relationship between the concepts is identified and annotated as generic "association" as shown in Fig. 13.

5. PULL FUNCTION

It is important to notice that our system allows the user to get the information directly without referring to any text. Practically, it means that we have several modes for results: text referring results and data returning results. The latter mode works like data retrieval in a database system. The user can specify a concept or a concept with relationships. Our system can return several types of results: text, text fragment(s) containing the concept or frequency of occurrence of the concept in the text.

Figure 11. Another example of annotations of concept hierarchy

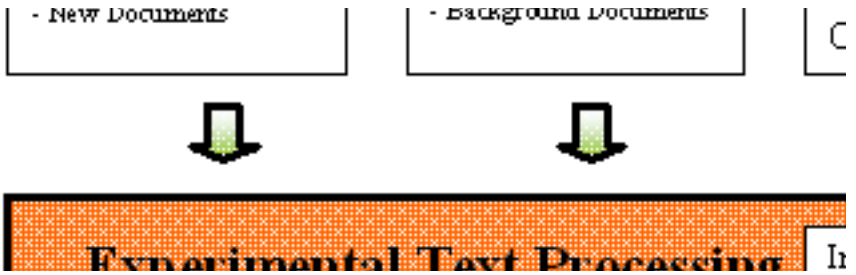
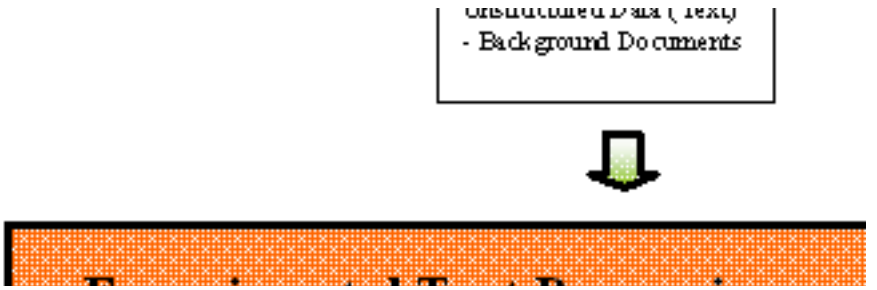


Figure 12. Annotation of relationships



Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 13. Document with relationships annotated

1. **Our**
2. <concept name="CASS organization" type=" compound " source= " ontology ">
3. < concept name="CASS" type="simple" source=" text"> > **CASS** </concept>
4. < concept name=" organization" type="simple" source= " ontology " hierarchy=" super-concept" >**Organization** </concept></concept>
5. < relationship name="association" type="simple" source="text" > **is involved in a variety of** </ relationship>
6. < concept name="study" source="ontology" type="simple"" source= " ontology " > **studies** </concept>
7. **.**

5.1 Keyword Search

Most of the existing text retrieval system have this feature implemented. Here the queries are based on simple keywords or any logical combinations using OR, AND, and NOT operators. Typically the documents are ranked based on these keywords and the most relevant documents are returned. Some text processing system also contain a text mining module that can cluster documents based on concepts.

5.2 Queries Involving Concept Hierarchy and Concept Relationships

Concept hierarchy and relationship annotations allow the user to issue specific data queries e.g. "Give me the names for the projects done by CASS" for a set of documents. For our example, the result would be "disease rates". A document fragment also could be displayed to show this information. Some statistical ranking and clustering can be improved when text is annotated with concept hierarchies and relationships.

6. PUSH FUNCTION

After the new document is processed, there are three types of information that can be "pushed" by the system: (a) "new document is consistent with background document," (b) "new document is inconsistent with background document," or (c) "new document is disjointed with the background document".

For the military medical documents, the most important situation is to identify the case when "new document is inconsistent with background document." Therefore, our experimental system should trigger some action, in this case, e.g.: alert subject matter expert, about the situation who make the makes the final decision such as review and updates the background document. For other business applications, when "new document disjoint with the background document," it may also be important to trigger some action.

The push function requires comparison of documents: the new document with background documents. The background documents indirectly provide the list of concepts of interest. There are many levels of operation for the Push module. At the first level, it compares concepts. This is very similar to the pull function operated at the simplest mode.

6.1 Push Function Based on Concepts

There are many levels of operation for the Push module. At the first level, it compares concepts. This is very similar to the pull function operated at the simplest mode. In this case, our system returns several types of results, e.g. pairs of text fragment(s) containing the same concept or pairs of frequencies of occurrence of the concept in the new and background text. It is possible to add keyword restriction if necessary. There is a need to experiment with the system and to tune it to satisfy the requirements. For example, it is possible by specifying the general constraints between concept frequency matrices (including collocated concept matrices). In general, however, that level may be appropriate to find disjointed documents but may be insufficient for "similar" documents. It typically would cluster together both "consistent" and "inconsistent" documents. Therefore, for our purpose, we need to include concept hierarchy and/or relationships in document comparison.

6.2 Push Function Based on Hierarchies and Relationships

Let us consider the following document fragment as a new document.

Battle injury (BI) rates are the main topic of CASS studies.

Figure 14. Pull and push module

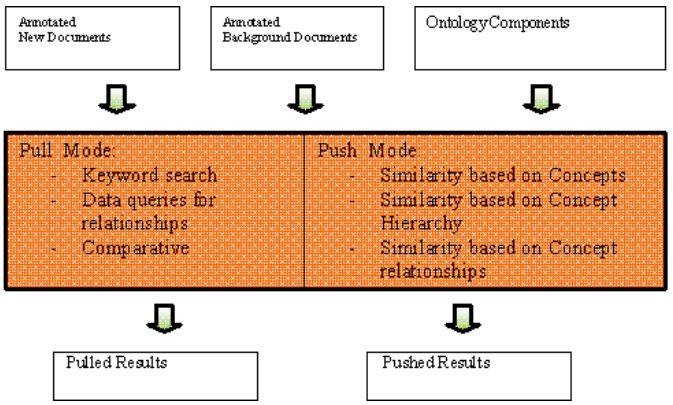


Figure 15. New document with concepts and relationships annotated

1. < concept name=" BI rate" type="compound" source="text" >
2. < concept name="BI" " type="compound" source="text" >
3. < concept name=" battle" " type="simple" source="text" > **Battle** </concept>
4. < concept name=" injury" " type="simple" source="ontology" " hierarchy=" super-concept" > **injury** </concept> **(BI)** </concept>
5. < concept name=" rate" " type="simple" source="ontology" " hierarchy=" super-concept" > **rates** </concept></concept>
6. < relationship name="association" type="simple" source="text" > **are the main topic of** </ relationship>
7. <concept name="CASS study" type=" compound " source= " ontology ">
8. < concept name="CASS" type="simple" source=" text"> > **CASS** </concept>
9. < concept name=" study" type="simple" source= " ontology " hierarchy=" super-concept" > **studies** </concept></concept>**.**

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Let us also assume that the previous text fragment given in Section 4.1 constitutes the background document. The provided above new document fragment can be processed by Text Annotation Module resulting in the document annotations as shown in Fig. 15.

Comparison of the concept hierarchies in new and background document would reveal the semantic differences much more precisely. We can automatically discover inconsistencies between the background information that claims that "CASS main study is disease rates" and the new document that claims that "CASS main study is battle injury rate" (please note the role of word *main* in the analysis).

There are many parameters that can be specified for hierarchy comparison functions that are used in the Push Module. For example, a comparison function can take into account the inheritance of relationships, e.g. if the background document contains "CASS main study is injury rates," then the new document "CASS main study is battle injury rates," would be accepted as "consistent" with the background document. The same comparison function can perform matching at various hierarchy levels.

In addition, hierarchy comparison functions or relationship comparison function can use equivalence transformation between "is_a" relationship and "super-concept."

In general, the use of a combination of matching functions may be required. Then we either need to define their priority, or establish some global measure of "inconsistency" based on measures of individual functions.

7. SUMMARY

The goal of this project is to provide an experimental processing system for unstructured data in the dynamically changing environment of new and background documents. The project allows for automatic generation of XML annotations and their use in retrieval systems. Our approach in this project is to use a combination of qualitative and quantitative technique of identification of basic concepts and relationships with respect to the domain of interest represented by a well established ontology. The immediate use of these techniques are for pull and push business functions.

NOTE

The views expressed in this article are those of the authors and do not reflect the views of the Army Medical Department, Department of the Army, or Department of Defense.

REFERENCES

[1] AltaVista Search Engine, http://www.altavista.com

[2] R. Bayardo et al. InfoSleuth: Agent-based Semantic Integration of Information in Open and Dynamic Environments. *In Proceedings of the ACM SIGMOD*, pages 195-206. ACM Press, 6 1997.

[3] E. Brill. Transformation-based Error-driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, vol. 21 pages 543-565, 1995

[4] B. D. Czejdo, and C. Sobaniec. Using a Semantic Model and XML for Document Annotation. To appear in *Proceedings of IEA/AIE-2000, Lecture Notes in Computer Science*, Springer-Verlag, 2000.

[5] J. Dinsmore. Layout, A Road Map for Information Extraction. Technical Report INSL-094-00, MCC, 1999

[6] J. Dinsmore. The Next Generation of Information Extractors. Technical Report SRI-068-99, MCC, 1999

[7] C. H. Hwang. Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information. Technical Report INSL-043-99, MCC, 1999

[8] Y. Labrou and T. Finin. Yahoo! As an Ontology – Using Yahoo! Categories to Describe Documents. *In Processing of CIKM*, pages 180-187, Kansas City MO 1999

[9] L. Liu et al. An XML-based Wrapper Generator for Web Information Extraction. *In Proceeding ACM SIGMOD*, pages 540-543. Philadelphia PA, 1999

[10] D. Mattox, L. Seligman, and K. Smith. Rapper: A Wrapper Generator with Linguistic Knowledge. *In Proceeding of WIDM*, pages 6-11. Kansas City MO, 1999

[11] M. Nodine et al. Active Information Gathering in InfoSleuth. *International Journal of Cooperative Information Systems*, 5(1/2), 2000

[12] World Wide Web Consortium, XML Home Page, http://www.w3.org/XML

[13] Yahoo, http://www.yahoo.com

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/proceeding-paper/pull-push-business-functions-experimental/33180

Related Content

Open Data Policy and Practice
Terry Buss (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 5188-5198).*
www.irma-international.org/chapter/open-data-policy-and-practice/112968

Designing a Concept-Mining Model for the Extraction of Medical Information in Spanish
Olga Acostaand César Aguilar (2021). *Encyclopedia of Information Science and Technology, Fifth Edition (pp. 856-872).*
www.irma-international.org/chapter/designing-a-concept-mining-model-for-the-extraction-of-medical-information-in-spanish/260234

Distributed Parameter Systems Control and Its Applications to Financial Engineering
Gerasimos Rigatosand Pierluigi Siano (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 15-35).*
www.irma-international.org/chapter/distributed-parameter-systems-control-and-its-applications-to-financial-engineering/183717

Dynamics in Strategic Alliances: A Theory on Interorganizational Learning and Knowledge Development
Peter Otto (2012). *International Journal of Information Technologies and Systems Approach (pp. 74-86).*
www.irma-international.org/article/dynamics-strategic-alliances/62029

Internet Memes
Lars Konzack (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 3770-3776).*
www.irma-international.org/chapter/internet-memes/112814