Information Retrieval and Processing–Setup of a Full Text System Implementing Automatic Metadata Extraction and Visualization

Bernie Huan, Chinweze Ubadigha, Dexter Chen, Eric Chang, Eric Lee, Feng-Chun Hsia, Henry Peng, Hoang Tan, I-Chieh Lin, Jacky Wu, Jim Lan, Jones Hou, Karthick Mani, Kenny Hsu, Piyarul Hoque, Rahul Aditya, Ray Chang, Tan Phat, Wei, Yu-cheng Chen, Kenvin Lo, Torbjörn E. M. Nordling

June 21, 2016

1 Introduction

This technical report contains both general information on the state of the scientific information retrieval and processing art and a description of the Nordron-SciInfo software package for information retrieval and processing. This report was written by all participants of the *Scientific Information Gathering and Processing for Engineering Research* course lead by Prof. Nordling at the National Cheng Kung University in the spring semester 2016. The main objectives of the course is to teach the students state of the art information retrieval and processing methods, project management, and technical writing by doing a project on implementation of some methods for retrieval and processing of full text scientific articles. The result of the student project is described in this report.

1.1 Aim and project structure

This section should be updated based on what you are building.

The whole project was separated into three parts, each with a responsible team:

1. Create a data base of open access full text articles (Team Wolverine).
2. Create a complementary XML like structure for metadata that contains all information in the PDF and can be show in Utopia (Team Eagle unit).
3. Automatic creation of metadata/markup by use of natural language processing of full text articles (Team Union).

2 Background

2.1 Information retrieval on existing database

Author:Dexter Chen, Eric Chang, Eric Lee, Jacky Wu, Karthick Mani, Kenvin Lo, Yu-cheng Chen.

We live in the time where technologies evolve beyond our imagination. Information growth in a exponential rate according to Tague et al. (1981), thus we can't rely on the old fashion ways to find data we want. We need new information retrieval methods to handle such a big amount of data systematically. But most of the information retrieval methods such as search engine can't really search everything on the web. They can only search the data that has been captured into the database according to Grehan (2002). Thus, we need to create a database to store these data and automatically update them frequently.

There are several online libraries currently available for us to get the academic articles or periodicals we need. And they can be roughly divided into three groups according to the way they store articles base on the division used by National Taiwan University Library.

Index libraries These kind of libraries stores the index and abstract of the articles. They don't provide the full-text documents directly, but may give the linkage to the publisher websites of articles.

And they can be categorized by the type of articles they include.

- **Comprehensive topics**
  Libraries such as Web of Science, S
- **Specialized topics**
  Libraries such as Compendex, BI views, PubMed, MEDline...

Good that you start by describing what it is and then tell how it is built.

Publisher libraries These libraries are created by the publishers themselves, so they provide the newest and complete documents directly. And can also be categorize by the type of articles they include.

- **Comprehensive topics**
  Libraries such as Science Direct, Springer Link, Wiley Online Library...
- **Specialized topics**
  Libraries such as Nature.com, Emerald Management Xtra, IEEE Xplore...

Aggregator libraries These libraries do not publish the articles by themselves, but they still sometimes provide the full-text articles to the user. The way they do this is to negotiate with some of the publisher libraries and get the authorization of the articles. Libraries such as EBSCOhost, ProQuest, JSTOR...

The comparison between three kinds of library can be found on Figure 1. On the next section we'll discuss about more details about some of the existing libraries.

2.1.1 Introduction to several libraries

Consider to change the title.

1. **PubMed** PMC (PubMed Central) is launched in 2000. and PubMed is a free library which is used for searching reference papers and abstracts related to the biomedical topics. THe design philosophy of PubMed is based on full-text XML files, which are readable by both the machines, humans and moreover technology independent. PubMed is classified into Index libraries, which is the prime reason that it is not able to provide full text in some papers. For the type of database used by PubMed is Microsoft SQL server, which is a relational database to store all of the archives such as XML, images, PDF files supplementary, etc.

   PubMed citations often include links to the full-text article on the publishers' Web sites and/or in PMC and the Bookshelf.

   MEDLINE is the largest subset of PubMed. You may limit your PubMed search retrieval to MEDLINE citations by restricting your search to the MeSH controlled vocabulary or by using the Journal Categories filter called MEDLINE.

   Simple searches on PubMed can be carried out by entering key aspects of a subject into PubMed's search window.

   PubMed translates this initial search formulation and automatically adds field names, relevant MeSH (Medical Subject Headings) terms, synonyms, Boolean operators, and 'nests' the resulting terms appropriately, enhancing the search formulation significantly, in particular by routinely combining (using the OR operator) textwords and MeSH terms.

2. **IEEE Xplore**

   IEEE Xplore is a scholarly research library formerly known as IEEE/IET Electronic Library (IEL). The IEEE is an acronym for Institute of Electrical and Electronics Engineers, which is one of the leading standard organizations in the world. More than 3.5-million full-text documents in the field of electrical, engineering, computer science and electronics are provided in this library.

   The front and user interface of IEEE library present the information on the screen, including the latest Angular, Jquery, HTML 5, CSS, etc.Most of the HTML for PDF, either it is for journal (conference) articles or standards get dynamic transformations real time and served through MarkLogic. Endeca, which is an Oracle product powers Xplore searches, is used in the search layer. All PDF files are fed through Endeca system. Endeca servers will provide the matching documents and Xplore platform presents it on the screen to the user. And all the content is stored in oracle metadata which will be consumed by Endeca, MarkLogic Authentication, and Authorization services.

3. **EBSCOhost**

   EBSCOhost is a popular reference which authorizes users to gain a great many full-text ar-

You mean it is implemented using these techniques. Please don't use etc. when you describe how something is built and do it in this context.

ticles from proprietary databases. EBSCO Information Services, headquartered in Ipswich, Massachusetts, which is a division of EBSCO Industries Inc., the third largest private company in Birmingham, Alabama, with annual sales of nearly 2 billion according to the BBJ's 2013 Book of Lists.

EBSCO offers library resources to customers in academic, medical, K–12, public library, law, corporate, and government markets. Its products include EBSCONET, a complete e-resource management system, and EBSCOhost, which supplies a fee-based online research service with 375 full-text databases, a collection of 600,000-plus ebooks, subject indexes, point-of-care medical references, and an array of historical digital archives.

In 2010, EBSCO introduced its EBSCO Discovery Service (EDS) to institutions, which allows searches of a portfolio of journals and magazines

4. **Comparison Xplore**

PubMed is a free library which contains many databases, like Medline, PreMedline and Publisher Supplied Citations. One can also access Medline through EBSCOhost. Medline is the largest subset of PubMed. You may limit your search to Medline only in PubMed. Both of them are built by National Library of Medicine. In contrast to other two libraries, the advanced search of PubMed is weak. It do not show citation times or further information. The documents in PubMed are almost related to the biomedical topics. IEEE contains more than one third documents in the field of electrical, engineering, computer science and electronics. And EBSCOhost PubMed is a free search engine accessing primarily the MEDLINE database of references and abstracts on life sciences and biomedical topics.

You need to add at least one library example of each library type, since you started with different types. Otherwise you could have focused on one type and motivated the focus. For the libraries containing articles you should add several since you are building one. You should also discuss the database techniques, including alternative ones to the used ones, such as MongoDB, Hadoop, etc. Try to compare features.

2.2 XML metadata structure

Author : Chinweze Ubadigha, Feng-Chun Hsia, Henry Peng, I-Chieh Lin, Jones Hou, Piyarul Hoque, Ray Chang.

To make the database functional for the user to receive the articles that they are looking for, our responsibility will be creating an interface between the user, the database, and the searching programme. In the other words, we're going to construct a webpage with a search bar for the user to enter their search string, And can display the search result given by the search system. For displaying the information of each article the system found, we need to construct an XML schema which contains all important data about the article.

This article provides an overview of metadata standards which are related to our responsibility. A number of metadata schemas in use to now-a-days are reviewed, including MODS, METS, METS+MODS+PREMIS, MARC 21, MARCXML, Dublin Core, and their pros and cons. Finally, we compare these schemas by examining the characteristics and unique features of them. We are able to rank them and suggest the optimum standard to build our XML structures. XML is a markup language that defines a set of rules for encoding the documents in a format, which is both human-readable and machine-readable. It is widely used to representation of arbitrary data structures, such as those used in web services.

Metadata is defined as the "data about data" or alternatively "information about information". In practice, metadata summarizes basic information of data for the organization and management of documents. It can be accessed manually or by automatic information processing and coding. Underwood and Watson (2003).

The metadata schemas can be classified into three types Dempsey and Computer (2015):

- Simple formats: it includes relatively unstructured data, typically automatically extracted from resources and indexed for searching. The data has little explicit semantics and does not support searching by field, such as Lycos, Altavista, Yahoo, etc.

- Structured formats: it includes data which contains the full enough description to allow a user to assess the potential utility or interest of a resource without having to retrieve it or connect to it. The data is structured and sup-

Comparison between three types of libraries		
Type	Advantages	Disadvantages
Index	1) Extensive coverage 2) Not limited to a particular journal publishers	1) Abstract only 2) Without full-text
Publisher	1) Easy to get full-text 2) Latest publications will upload very quickly	1) Journal from particular publisher only 2) Less of literature (limitation)
Aggregator	1) Can find full-text 2) Extensive coverage like Index libraries	1) Several full-text paper will embargo as authorize of publisher 2) Latest publications can not upload as quickly as Publisher libraries do

Figure 1: Comparison between three types of libraries.

ports fielded searching, such as Dublin Core, IAFA templates, RFC 1807, SOIF, LDIF.

- Rich formats: it includes fuller descriptive formats which may be used for location and discovery, but also have a role in documenting objects or very often, collections of objects, such as ICPSR, CIMI, EAD, TEI, MARC.

XML (Extensible Markup Language) is the universal format for the encoding and exchange of structured documents and data. There are no predefined tags and document structures in XML. In other words, the XML provides structural capabilities that HTML lacks, making it easy to achieve the principles of modularity and extensibility. The XML schema specification defines a schema language that allows for the specification of application profiles that will increase the prospects for interoperability Duval et al. (2002). Our work is to build a metadata schema in XML structure. The following sections will introduce the different schemas as mentioned previously.

Different standards of metadata

There are various types of standards that describe the metadata in different fields and applications. Listed in the following are five different standards with a brief introduction to each one.

1. METS
   Introduction

Metadata Encoding and Transmission Standard (METS) is an XML encoding format for storing the descriptive, administrative, structural and behavioral metadata needed to manage complex digital objects in an open and standardized way.

In 1990s, Making of America II (MOA2) project was proposed to share vision between national digital libraries which provides a mean for the Digital Library Federation (DLF) to investigate, refine, recommend metadata elements and encodings used to discover, display, and navigate digital archival objects. MOA2 DTD was created to test MOA2 project.

However, MOA2 DTD was limited in several ways. It provided no flexibility in terms of the exact metadata elements to be used for descriptive, administrative and structural metadata. Also, limited in scope to support for text and still image materials and no attempt to support time-based media such as audio or video materials. In order to solve those problems led to the creation of METS.

Advantages

(a) METS to facilitate the exchange and interoperability of digital library objects across digital library systems.

(b) Provide and support a practical and flexible packaging mechanism for the long-

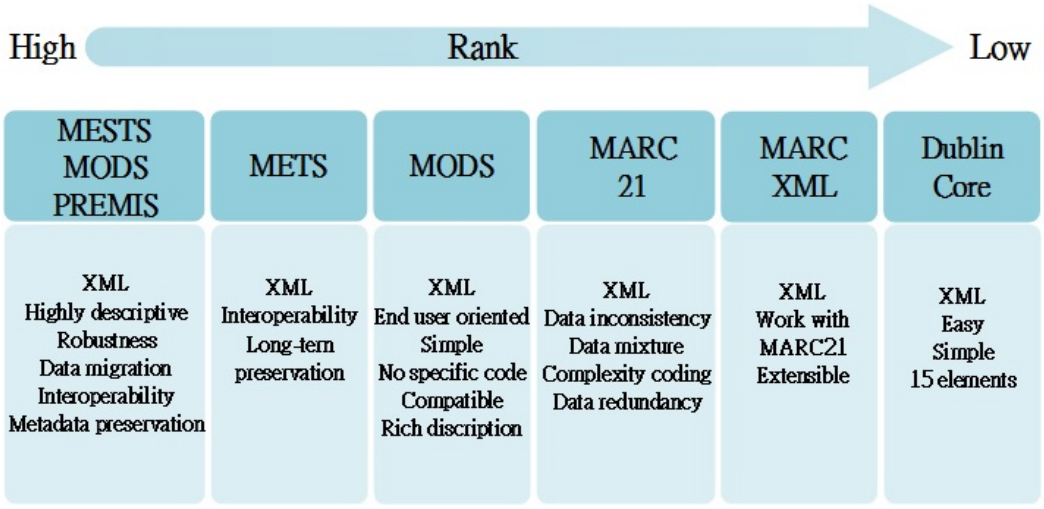

Figure 2: Overview and hierarchical ranking of metadata standards and their individual features

term preservation of digital library objects.

(c) The METS standard can be considered as one of the many efforts to try to determine, for one particular community, how complex sets of data and metadata might best be encoded to support both information exchange and information longevity.

Disadvantages

(a) METS has gone some distance towards achieving these design goals, it is not itself in a guarantee of interoperability.

(b) There are some obvious practical difficulties in using METS for the long-term preservation of digital objects.

Conclusion

2. MODS
Introduction
Metadata Object Description Schema (MODS) was developed by the Library of Congress' Network Development and MARC Standards Office in 2002. It is the bibliographic element set for multiple purposes, which was especially for library applications. As an XML schema, it is not only able to carry the selected data from existing MARC 21 records but to enable the creation of original resource description records. It includes a subset of MARC and uses language-based tags rather than numeric ones. In some cases regrouping elements are from the MARC 21 bibliographic format. It released the third version (version 3.6) in May 2015. MODS is expressed using the XML of the World Wide Web Consortium. The standard is maintained by the MODS Editorial Committee with support from the Network Development and MARC Standards Office of the Library of Congress.

MODS is an XML schema which is guidelines a resource description for encoding, as well as exchange and management descriptions of encoding.

Elements of MODS generally inherit the MARC, some data has been repackaged; in the some cases what is in several data elements in MARC may be brought together into one in MODS. Also, MODS does not assume any specific cataloging code.
It is used as an extension schema to METS (Metadata Encoding and Transmission Standard), as a representing a simplified MARC record in XML.

Advantages

(a) The element set is richer and more descriptive than Dublin Core.

(b) The element set is more compatible with library data than ONIX.

(c) The schema is more end user oriented than the full MARCXML schema.

(d) The element set is simpler than the full MARC format.

ONIX: ONIX is an XML-based standard for rich book metadata, providing a consistent way for publishers, retailers and their supply chain partners to communicate rich information about their products.

Disadvantages

(a) An original MARC 21 record converted to MODS may not convert back to MARC 21 in its entirety without some loss of specificity in tagging or loss of data.

(b) In some cases if reconverted into MARC 21, the data may not be placed in exactly the same field that it started in because a MARC field may have been mapped to a more general one in MODS.

(c) MODS does not include business rules for populating the elements.

(d) Additional instructions would need to be provided for conversion details.

Conclusion
MODS has a high level of compatibility with MARC records because it inherits the semantics of the equivalent data elements in the MARC 21 bibliographic format. It may be used for the original resource description that allows for rich description that is generally compatible with existing library data and is expressed in XML syntax. Because it includes a subset of MARC fields and repackages some of them, it is particularly useful for technician input.
An additional use of MODS is as an extension schema for descriptive metadata for the METS object.

3. METS+MODS+PREMIS
Introduction
The first digital repositories was developed by British Library's e-journal system which combined METS, MODS and PREMIS.Dappert and Enders (2008) The system took the advantage of the METS structural, PREMIS preservation and MODS descriptive metadata to form a advanced metadata structure. The Metadata Encoding and Transmission Standard (METS) is an XML document that can package the metadata of a digital resource: the descriptive, administrative, structural, rights and other data needed for retrieval and preserving of a digital resources.Guenther and McCallum (2003) In other words it can be referred as a metadata storing and communication standard. The METS wrapper has up to seven major subsections: "a METS Header (metsHDR), a Descriptive Metadata Section (dmdSec), an Administrative Metadata Section (amdSec), a File Section (fileSec), a Structural Map (structMap), Structural Links (structLink), and a Behavior Section (behaviorSec)" these form the basic structure of METS. The Structural Map is the most important subsection and must be included in a METS document.Cheslow (2014) These subsections have elements that provide the means for describing in detail the digital objects. The Structural Map defines a hierarchical structure such that using METS pointers users of the digital library object can easily navigate through it. One great advantage of METS is that it provides a flexible framework for modelling different document types and scenarios.Dappert and Enders (2008) The Metadata Object Description Standard (MODS) provides ways to describe objects and has a high level compatibility with MARC. Among other XML metadata standard it is an alternative between a simple metadata format (such as Dublin Core) which has a minimum of fields and little or no substructure, and a very detailed format (such as MARC 21) with many data elements having various structural complexities.Guenther and McCallum (2003) The PREservation Metadata Implementation Strategies (PREMIS) is an administrative metadata schema used for the preservation of digital resources.Cheslow (2014) With the rapid changes in technology, digital objects including its metadata are bound to go obsolete at some time in the future. PREMIS was created to set standards that will ensure long term usability and preservation of digital resources.

Why METS+PREMIS+MODS?
Understanding metadata needs, which is important to duscuss the data production and structures. Structuring digital objects particularly e-journals present two main difficult problems. First, e-journals are structurally complex. New issues are released in intervals for each journal title. These may contain a varying number of articles and other publishing matters having a variety of formats. Second, the production of e-journals are outside the control of the digital repository and done without the benefit of standards for the structure of file formats, metadata formats and vocabulary, publishing schedules, etc.Dappert and Enders (2008) As a means to solve these problem, METS provides a robust and flexible way to define digital objects. The MODS on the other hand, provides ways to describe digital objects and can be built on a MARC. Finally the PREMIS provides ways to describe digital objects and processes that are essential for digital preservation. Also, these three metadata standards are all built on an XML schema.Dappert and Enders (2008) Details on how to implement these three metadata standards to form a robust metadata structure or archive can be found in "Using METS, PREMIS and MODS for Archiving eJournals".Dappert and Enders (2008) Though there are different ways to implement these schema only one was discussed in the aforementioned literature.

Advantages

(a) Interoperability: According to Hafezi et al on their survey of Iranian digital library, most of the bibliographic data comprises of 82% XML and 64% MARC formats. Given these statistics, METS+PREMIS+MODS can be considered interoperable since they all can be implemented in these formats.Alipour Hafezi et al. (2013)

(b) XML Schema: Considering our given responsibility and the easiness of implementing XML, METS+PREMIS+MODS is among the right choice.

(c) Metadata Preservation: The inclusion of PREMIS in METS provided the metadata preservation feature which single metadata standard cannot provide.

(d) Highly descriptive metadata: The MODS used in structuring the descriptive metadata in METS provided a highly descriptive metadata structure.

(e) Data migration: Because METS is flexible and contains header for easy transmission it is very easy to deploy this metadata structure to a different system.

(f) Robustness: This schema is considered robust by the virtue of containing the features of three different metadata standard.

Disadvantages

(a) Easiness: This schema is not ease to build compared to single metadata structure.

(b) Redundancy: Some of the metadata stored in the METS were also stored in the PREMIS to improve preservation.

(c) Update: The digital object in the repository are write-once in order to support archival authenticity and track digital object provenance, thus in-situ update is not possible. To update another version of the Archival information package has to be added.

Conclusion
METS is an excellent metadata schema for use with digital libraries and will become more robust when combined with MODS for descriptive metadata and PREMIS for preservation metadata. Also, with a minimal knowledge of XML, METS is relatively easy to implement and the Library of Congress provides great resources to help implement METS. Our mission is to create a better metadata structure that can stand the test of time. Bearing this in mind and considering the metadata standards mentioned above, the combination of METS, MODS and PREMIS possesses the features that will resolve the limitations of present day information retrieval systems.

4. MARC 21

Introduction
The Library of Congress Network Development and MARC Standards is developed a framework for working in MARC data in a

XML environment. The MARC XML schema does not need to be edited to reflect of minor changes to MARC 21. The schema retains the semantics of MARC.
This information has been made in several areas and fields, one of these is the bibliographic domain, where it is guided by instruments, principles, models, and technologies. With the metadata standards used in this field, the MARC formats 21, with origins in the 1960s. Considering the widespread use of these standards are, the objective of highlight the purposes that led to the creation of MARC21 formats. Which carried out a literature review on the origin of MARC and its development to the MARC21 and the coding records. Thus, it is presented coding with XML and the MARCXML schema, as well as criticism of the MARC21 formats. It follows that, despite the criticism, the MARC formats 21 are still used and disseminated, and despite the advantages offered by XML, with the ISO 2709 standard.
It is important to know that the MARC 21 is a data exchange format, which tells how import or export successfully occur the cataloging record and bibliographic and should be described. But the catalog data model should not necessarily be structurally organized in the same format as a MARC21 record.
When the technological development starts from direct and indirect implications for informational resource representation exchange of cataloging data and activities. We hoped that the MARC21 formats, on their encodings and development have contributed to the area of Information Science.
But now-a-day the standards are the Metadata Object Description Schema (MODS) (Metadata scheme for description of object). And the Metadata Authority Description Schema (MADS), both created for use with XML and specified by XML schemas.

Advantages

(a) Data inconsistency: The same type of data is recording in different fields or subfields of different forms.

(b) Data redundancy: The same data is recording in more than one field or subfield, sometimes as a coded way and sometimes literally.

(c) Data mixture and their attributes.

(d) The coding is extreme complexity.

Disadvantages

(a) Problems due to shared cataloging environment for which MARC 21 was designed.

(b) Problems caused or partially caused by MARC 21 and that perhaps can be solved in the data migration process to a new standard of data structure in the future.

Validation of MARC 21 data

(a) Basic XML validation according to the MARC XML schema.

(b) Validation of MARC 21 tagging (field and subfield).

(c) Validation of MARC record content, e.g., coded values, dates, and times.

Conclusion
The MARC formats 21 are still used and disseminated for the exchange of cataloging data in digital environment. Despite the advantages offered by the coding XML, including the development of software for processing MARC 21 records still persists.
Together with efforts to use XML - coding in MARC21 records, LC is designed meta data standards which have alternatives to traditional formats. Among these standards are the Metadata Object Description Schema (MODS) (Metadata Scheme for description of object) and the Metadata Authority Description Schema (MADS), both created for use with XML and specified by XML schemas.
The MODS and MADS have great compatibility with the traditional formats of MARC 21, although in general do not allow the data record with the same level of specificity given by the MARC formats 21.
The MODS, due to the high compatibility with the MARC format 21 for Bibliographic Data can be chosen by libraries as a metadata standard for describing information resources.

5. MARCXML
Introduction
To make up the less of internet compatibility of MARC 21, the Library of Congress developed an XML schema based on it, which the schema is the MARCXML standard.

The purpose of MARCXML is to build a metadata format with a simple, extensible and flexible structure, which can be presented in XML stylesheets. Since MARCXML was designed to converge data from MARC 21, the structure and performance are pretty similar between these two standards.

Advantages

(a) Used in XML directly.

(b) Easily work with MARC 21 system.

Disadvantage

(a) The disadvantage of MARC 21 can be almost totally found on MARCXML, except the ability of internet application.

Conclusion
Since our responsibility is to construct an XML schema, MARCXML will be a good choice if MARC 21 becomes the standard to be worked with.

6. Dublin Core
Introduction
Dublin Core provides very simple but efficient sets of metadata. Dublin Core's four main principles are high flexibility, clear and easy to understand gereral connotation, global, and easy to produce or maintain. There are fifteen core elements. These simple elements can be further defined to generate more detailed metadata.
The original Dublin Core metadata element sets are as follow: 1. Title 2. Creator 3. Subject 4. Description 5. Publisher 6. Contributor 7. Date 8. Type 9. Format 10. Identifier 11. Source 12. Language 13. Relation 14. Coverage 15. Rights. (NISO)

Advantages

(a) Encourage authors and publishers to provide Metadata in the type that can be automatically collected by resource discovery tools.

(b) Encourage web publishing tool that contains element of the Metadata module to be founded, which further simplify the creation of Metadata records.

(c) DC records can be the basis of more detailed cataloging records.

(d) After the DC becomes the standard, Metadata records can be understood by the user.

Disadvantages

(a) There are no cataloging rules that determine how data will be filled in. So if I write " Contributor: Sam Smith ", I can also write "contributor = Smith, Sam.". It means that there is no consistency across different uses of Dublin Core.

(b) Does'nt work well for data conversion.

(c) Data values in non-mappable space will be left out, especially when a source schema has a richer structure than the target schema, e.g. from METS to Dublin core.

Conclusion
Because there are no cataloging rules, it makes Dublin core easy to use by anyone. On the other hand, this is something that goes against the article cataloging.

2.3 Automatic creation of metadata

Author:Bernie Huan, Jim Lan, Hoang Tan, Kenny Hsu, Rahul Aditya, Tan Phat, Wei.

We are producing a program that automatically generate and extract metadata with natural language processing. We also strive to generate XML files with metadata extracted. In the best scenario, we will even try to create a search engine together with other groups. Also, creating a sutiable interface and structure for users is necessary. Following discussion is our literature review on natural language processing.

Language understanding

Natural language understanding (NLU) is a subtopic of natural language processing in artificial intelligence that deals with machine reading comprehension, it's considered an AI-hard problem.

When users search for a sentence, how does the program understand certain inputs of text?

We could build a natural language understanding (NLU) system, in which the system's rules for semantic interpretation are learnt automatically from training data, which uses a set of possible yes-no questions that can be applied to data items. After that, it follows rules for selecting the best question at any node on the basis of training data by using a method for pruning trees to prevent over-training.

Name Recognition

The results could be a country or an animal if users search for the word, Turkey. The type (meaning instead of type) is very different.

There are a lot of misunderstandings like this if users search some words which have multiply meanings. Sometimes, the results are totally unrelated and this situation is always annoying. That would be troublesome when we count frequency of certain words to rank them.

Therefore, it is significantly crucial for a program to totally understand what users want by name recognition in natural language processing. The users can find out the results much quicker and can't get misunderstood (and will not be confused?).

The method to improve the problem above is "categorize the words based on different subjects,topic or genres" by using online database. Metadata is limited in digital libraries and web resources, try to enlarge them with meaningful, organized and desired categories Kules et al. (2006).

Besides dealing with mutiple-meaning words,the most important part of name recognition is to recognize the special names and terms such as locations,people name,even company names and academic terms. Therefore, it is better for search engine to know what user want and huge name corpus are necessary. Plus, this work also can assist previous work.

With above effort,users' exploration and overviews of information could be better supported. It will be very convenient to find the results we want and lower the possibility of misunderstanding if users are not very familiar with finding the appropriate result in specific fields.

Tun Thura Thet et al. (2010) Users don not have to filter the results which are ranked by browsing frequency popularity. Users just can obtain the information and relevance by clicking the specific categories. Also, users are able to choose multiply fields if the results include a lot of relevant fields. That's a big motivation for people to handle this problems.

A lot of online services have done similar tasks before. Thus,creating and using an online database or automated metadata creation are to be recommended.The reason is there are many advantages, including integrating with the other cloud services or scaling with what users need such as how to categorize the categories. It is beneficial for people who would like to create a convenient and personalized database or metadata.

Part-of-speech

In the English language, we can consider words as the smallest elements that have distinctive meanings. Based on their use and functions, words are categorized into several parts of speech, and the 8 major parts of speech in English grammar are: noun, pronoun, verb, adverb, adjective, conjunction, preposition, and interjection.

- Noun (names)
  A word or lexical item denoting any abstract (abstract noun: e.g. home) or concrete entity (concrete noun: e.g. house); a person (doctor, Jim), place (farm, Taiwan), thing (earring, refrigerator), idea (happiness), or quality (ambition). Nouns can also be classified as counted

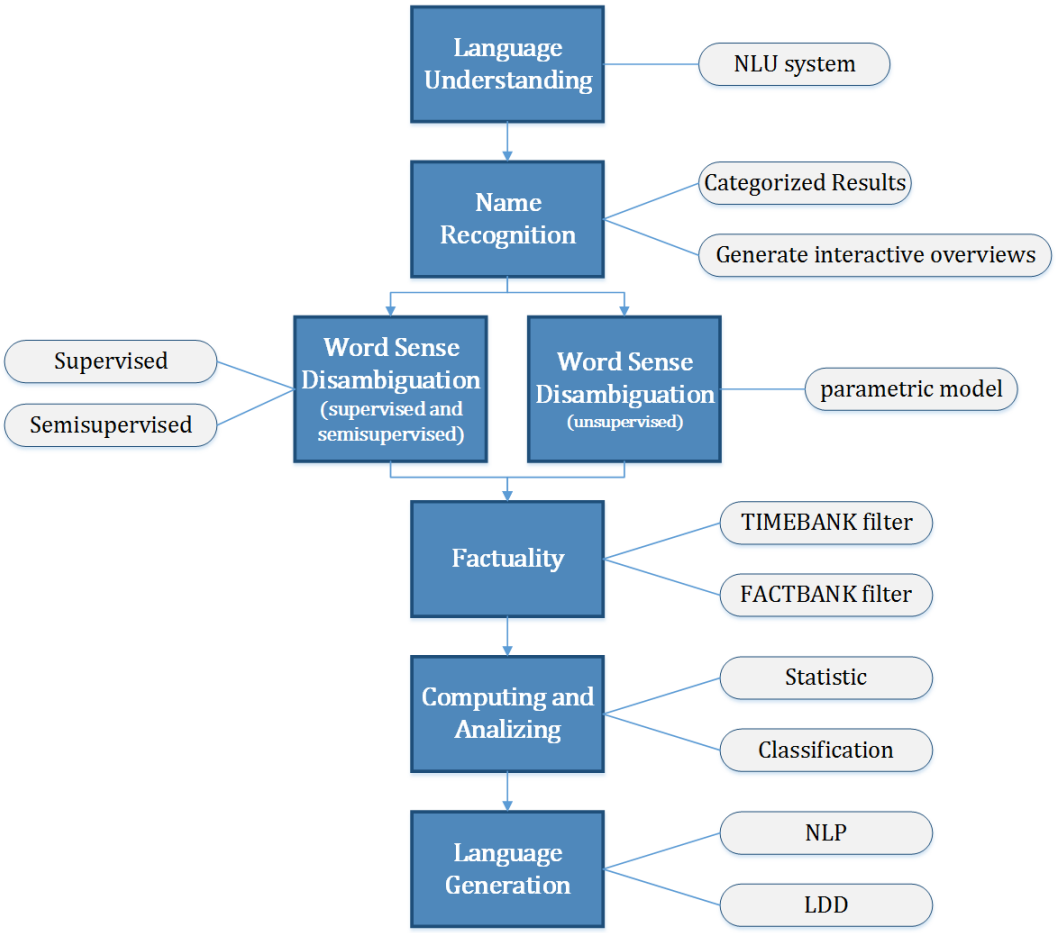

Figure 3: The process of metadata creation.

nouns or non-counted nouns; some can belong to either category. The most common part of the speech; they are called naming words.

- Pronoun (replaces)
  A substitute for a noun or noun phrase (e.g. them, he). Pronouns make sentences shorter and clearer since they replace nouns.

- Adjective (describes, limits)
  A modifier of a noun or pronoun (big, brave). Adjectives make the meaning of another word (noun) more precisely.

- Verb (states action or being)
  A word denote an action (walk), occurrence (happen), or state of being (be). Without a verb, a group of words cannot be a clause or sentence.

- Adverb (describes, limits)
  A modifier of an adjective, verb, or other adverb (very, quite). Adverbs make your writing more precisely.

- Preposition (relates)
  A word that relates words to each other in a phrase or a sentence and aids in syntactic context (in, of). Prepositions show the relationship between a noun or a pronoun with another word in a sentence.

- Conjunction (connects)
  A syntactic connector; links words, phrases, or clauses (and, but). Conjunctions connect words or group of words.

- Interjection (expresses feelings and emotions)
  An emotional greeting or exclamation (Huz-

zah, Alas). Interjections express strong feelings and emotions.

Part-of-speech tagging

Part of speech tagging (POS tagging), also called grammatical tagging or word-category disambiguation, which is a process of assigning a part of speech to each word in a sentence that based on both definition and its context.

Word sense disambiguation: supervised and semi-supervised approach

Word sense disambiguation (WSD) is an open problem of natural language processing and ontology. WSD identifies which sense of a word (i.e.meaning) is used in a sentence, when the word has multiple meanings Du et al. (2013).

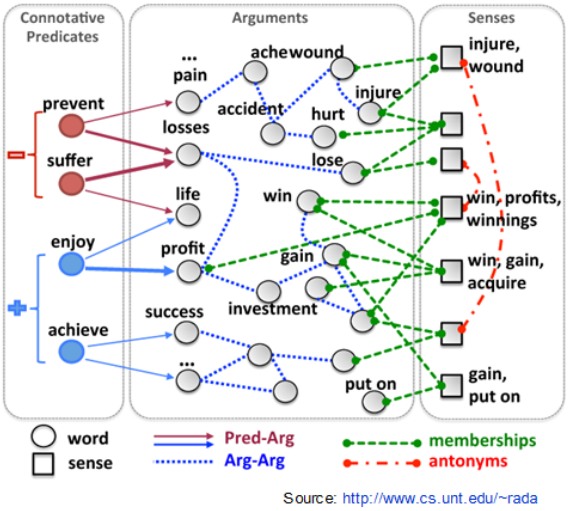

Figure 4: GWord+Sense with words and senses.

The solution to this problem impacts other computer-related writing, such as discourse, improving relevance of search engines, anaphora resolution, coherence, inference et cetera.

Word Sense Disambiguation(WSD) is related to Natural Language Processing and is also linked with computational languages. People introduced it as a solution when they felt the need of some complex problems like machine translation, information retrieval, speech processing and text processing ,etc.

WSD is mainly focused on determining the sense of word,computationally which is used in a problem by using that word in a particular context. Inspite of having a greater number of existing disambiguation algorithms, WSD still has an open problem with the three main parts of the WSD methods being considered by literature: Supervised, Unsupervised and semi-supervised.

The human brain is quite proficient at word-sense disambiguation. The fact that natural language is formed in a way that requires so much of it is a reflection of that neurological reality. In other words, human language developed in a way that reflects (and also has helped to shape) the innate ability provided by the brain's neural networks.

In computer science and the information technology that it enables, it has been a long-term challenge to develop the ability in computers to do natural language processing and machine learning.

Supervised

Supervised methods are based on the assumption that the context can provide enough evidence on its own to disambiguate words. Probably every machine learning algorithm going has been applied to WSD, including associated techniques, such as feature selection, parameter optimization, and ensemble learning.

Sense Tagged Text

Bonnie and Clyde are two really famous criminals, I think they were **bank/1** robbers
My **bank/1** charges too much for an overdraft.
I went to the **bank/1** to deposit my check and get a new ATM card.
The University of Minnesota has an East and a West **Bank/2** campus right on the Mississippi River.
My grandfather planted his pole in the **bank/2** and got a great big catfish!
The **bank/2** is pretty muddy, I can't walk there.

Source: http://www.cs.unt.edu/~rada

Figure 5: Sense tagged text.

Support Vector Machines and memory-based learning have been shown to be the most successful approaches, to date, probably because they can cope with the high-dimensionality of the feature space.

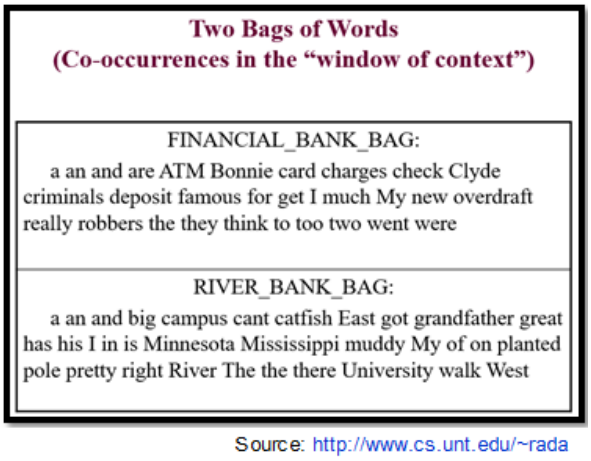

Figure 6: Two bags of words.

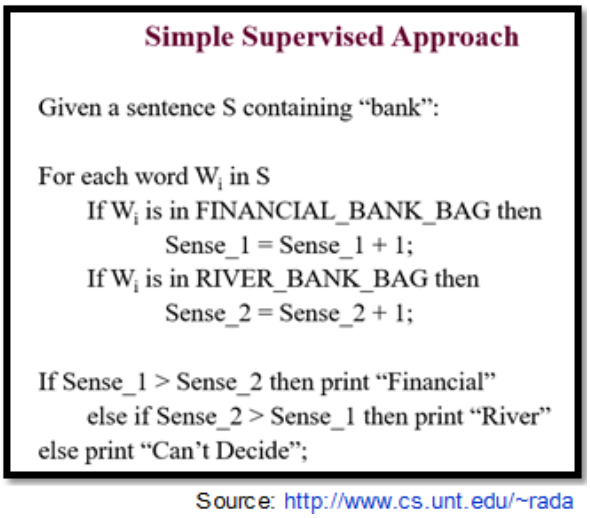

Figure 7: Supervised approach.

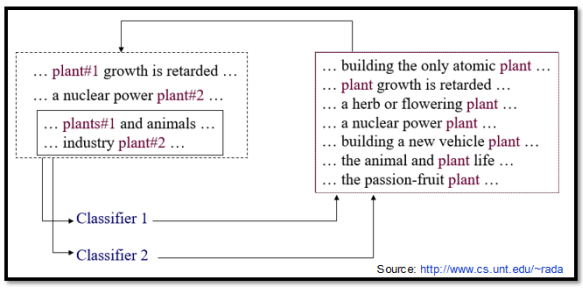

Figure 8: Classifier that improves over the basic classifier.

However, these supervised methods are subject to a new knowledge acquisition bottleneck since they rely on substantial amounts of manually sense-tagged corpora for training, which are laborious and expensive to create Ramos-Soto et al. (2015).

Semi-supervised

Because of the lack of training data, many word sense disambiguation algorithms use semi-supervised learning, which allows both labeled and unlabeled data. The Yarowsky algorithm was an early example of such an algorithm Gartner (2013). It uses the 'One sense per collocation' and the 'One sense per discourse' properties of human languages for word sense disambiguation. Based on observation it has been shown that words tend to exhibit only one sense in most given discourse and in a given collocation. Guo et al. (2010)

The bootstrapping approach starts from a small amount of seed data for each word: either manually tagged training examples or a small number of surefire decision rules. The seeds are used to train an initial classifier, using any supervised method. This classifier is then used on the untagged portion of the corpus to extract a larger training set, in which only the most confident classifications are included. The process repeats, each new classifier being trained on a successively larger training corpus, until the whole corpus is consumed, or until a given maximum number of iterations is reached Blascheck et al. (2016).

Other semi-supervised techniques use large quantities of untagged corpora to provide co-occurrence information that supplys the tagged corpora. These techniques have the potential to help in the adaptation of supervised models to different domains.

Also, an ambiguous word in one language is often translated into different words in a second language depending on the sense of the word. Word-aligned bilingual corpora have been used to infer cross-lingual sense distinctions, a kind of semi-supervised systemCheslow (2014).

Unsupervised

Unsupervised WSD is the third part of Word sense disambiguation,it has to focus in the sense of the word which is being used in a sentence. Unsupervised WSD, which rely on single writing can

be approached by the use of Naive Bayes' model, which mainly focuses on unsupervised part of the context. In this model, a number of sentences are used which contains a particular word which has several meanings. The main goal is to divide those words into a specified number of sense groups f. Wang et al. (2006). The Naive Bayes model applied mathematically entirely focuses on the issue of feature selection, which describes its two types:

1. Pedersen and Bruce local type features.
2. WordNet-based feature selection.

1. Pedersen and Bruce local type features:

Three different feature sets have been used by 'Pedersen and Bruce' (under Naive Bayes model) for each word to formulate such a model describing the distributions of sense groups of that word in Unsupervised WSD.

Features which were taken into account are: **Morphology:** The pattern of word formation in a particular language is called as Morphology. This feature represents the morphology Sak et al. (2010) of the ambiguous word and is denoted by M. In case of nouns, M acts as Binary which indicates whether the word is plural or singular. For verbs, M indicates the tense of the verb and can have up to seven possible values. This feature is not applicable for adjectives.

Part-of-speech: This feature represents the part-of-speech Creţulescu et al. (2014) of the word and tells the position of the ambiguous word. Each POS feature can have one of five possible values: noun, verb, adjective, adverb or other.

Co-occurences: This feature also acts as binary variables representing whether the most frequent content word in all the sentences contains the ambiguous word can occur anywhere in the sentence or not.

1. WordNet-based feature selection:

WordNet is a large lexical database of English. The approach to WSD relies on a set of features formed by the actual words occuring near the target word and reduces the size of this feature set by performing knowledge-based feature selection that relies entirely on WordNet. The WN semantic network provides the words considered relevant for the set of senses taken into consideration corresponding to the target word.

In WordNet, noun is the most developed portion as per the research done over the performance for knowledge-based disambiguation. For adjectives, the same disambiguation method has taken into account as the similarity relation, which is typical of this part of speech. Verbs are suggested to use additionally whenever possible. As a result of using only those words indicated as being relevant and recommended by WordNet, a much small vocabulary was obtained. and therefore a much smaller number of features were taking part in the disambiguation process.

So, this background focuses mainly on the issues of feature selection for unsupervised WSD performed with an underlying Naive Bayes model.

Really? This section is shorter.

The difference between 'Supervised' and 'Unsupervised' WSD is:

1. The Supervised WSD approach requires a large amount of data in order to achieve a reliable result and generally the scope is limited to some words. Whereas the Unsupervised WSD approach does not use any corpus and suggests the suitable information extracted to the word knowledge base. This method is used in case of performing WSD without data learning.
2. The Supervised approaches make use of information from labeled training data while the Unsupervised does not depend upon any labeled data, it uses a multi-lingual thesaurus that contains millions of biomedical and health related concepts, their synonymous names and their relationships.

A summary on list format can be motivated, but then each item need to be brief and you should not introduce anything new like UMLS in it.

Factuality

In the process of producing metadata, which should be the most precise information and representing the text, validity of such metadata must be checked. Therefore tools for fact checks are developed based on linguistic techniques.

The tool could detect facts and excludes authors' subjective opinions Agerri et al. (2014). From the authors's perspective, the two main set of tools having such functions is TIMEBANK and FACTBANK.(yes, the authors used capitalized name)

TIMEBANK was first proposed in Pustejovsky et al. (2003). The idea was based on that English language has different tenses which could be exploited as signals for fact check. An example below could help to clarify the ideas. Let's examine these sentences:

- I will go to Chimei museum tomorrow.
- Chimei museum is near Tainan District.
- I was in UK in 2012.

The first sentence is simple future tense which implies something has never actually happened, the second sentence is simple present tense which can directly imply facts, and the last sentence is in simple past tense which is about something already happened (which is facts), but is no longer a fact right now, so such fact must be used with caution.

The reason for introducing such tool is that even scientific research articles can be glittering with subjective comments, opinions or even assumption from authors Schultze (2000). In addition to TIMEBANK, many other tools can be another filter for fact extraction. Dave et al. (2003) Identify words, clauses and phrases that show emotional state of the authors.

The choice in expression of facts could also be a helpful indicator to show whether authors are subjectively supporting a cause, an opinion and so on Wiebe et al. (2005). Among these mentioned approaches, this paper highly favors creation a kind of thesaurus compiled of linguistic signaling for non-factually statements such as FACTBANK, which is built by Saur?? and Pustejovsky (2009). Following example shows how subjective statements can be picked out.

- Channelization would guarantee high flow velocity in rivers, flooding and consequent degradation of riparian community (1a)
- Funding agencies would be happy with big entrepreneurs, instead of small and medium enterprises (1b)
- Tolerance to dictatorship would has negative influences on anarchist movement (2a)
- Tolerance to dictatorship would doom anarchist movement (2b)

It is easy to find in statement (1a) is an absolute fact. Statement (1b) is however affected by emotional state of authors. After re-writing (1b) into: Funding agencies lend more money with lower interest rate to big entrepreneurs, instead of small and medium enterprises, sentence (1b) become a face-based statement. In another case, statement (2a) is a fact-based statement while in statement (2b), authors are stressing their dislike toward dictatorship.

Fact checks in language generation is a new field but many useful tools have been developed. Each of them has their own function and could complement each others. In the limit of this study, we are using both of TIMEBANK and FACTBANK together for fact check.

Language generation

Natural language generation (NLG) is one branch of natural language processing. The goal is generating the words human being using via machine automatically. To use this technique, six basic activities are done:

1. content determination: In this active, we create some messages which are communicated in the text. These messages shall be labeled and the entity in the messeges is also distinguished, which is convenient for us to use these data within the following step.
2. discourse planning: This part is closely related to the previous part. We determine the order and the structure of the messeges.
3. sentence aggregation:This part combine several messeges into sentence. Although some of messeges have been a sentence, we can improve the influency of messeges by combining them.
4. lexicalization: This part make the messege more precise by use the specific words and concepts. Then people can get the idea of the messeges more quickly.
5. referring expression generation: This part is a litte same as the previous part. But the difference is that this part differentiate the one domain to the other domains.
6. linguistic realization: Last part is to make the expression follow the rules of grammer, part of speech and the natural language rule.

Ramos-Soto et al. (2015).The advantage of this technique is that it is flexible, since there is no

Where? Please explain the 6 terms. What are they?

standardization. But it also has the difficulty in the implication of this technique.Ramos-Soto et al. (2015)No standardization means that no rules can be followed.

Without logic method, It is almost impossible to code and be realized by computer. Thus, the another concept is proposed. This way has the logical method, also the algorithm is easy to realize. The technique is linguistic discription of data. Linguistic description of data (LDD) is a concept that applied the fuzzy set theory in the linguistic field. At the beginning, comparing to the NLG field, it is a newer technique to solve the problem of language generation. However, the basic steps of LDD have been built. The four main parts in this technique are input data, linguistic variable, fuzzy quantifiers and evaluation criteria Ramos-Soto et al. (2015). Some of them are similar to the concept. The advantage of this technique is that it has been implied in many fields just like weather forecast Tamine et al. (2009). Also, many practical methods have been proposed. However, it still has a long way to go.

Now you jump too far. More explanations are needed.

These two techniques are usually combined together nowadays. The concept of NLG and the practical approach of LDD could be used in the same time to provide the better performance in language generate field.

3 Methods

3.1 Built a database containing ten thousand articles

The most important thing on this subject is to build a database containing 10,000 articles. To reach that target, this study will go to discuss the advantages and disadvantages of different databases and will decide the most suitable one to be the database of this study. At a meantime, this study will focus on how to use web crawlers to download articles automatically, which will be contained in the database of this study.

3.1.1 Database Management Systems

A database is an organized collection of data. A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. Data management comprises all the disciplines related to managing data as a valuable resource. It is the collection of schemas, tables, queries, reports, views and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

The relational database model was proposed by Edgar Codd in 1970. However, it was not universal at that time because of the technical requirements. Until the 1980s, first commercial relational database management system(RDBMS) which is the most popular database management system(DBMS) at present began to appear. Besides RDBMSs, there are several kinds of DBMSs. For example, object-oriented databases(OODBMS) and graph database management systems(GDBMS). In accordance with the definition, a database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. Well-known DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase and IBM DB2. Furthermore,they can support different kinds of databases.

1. **Object-oriented database**

   An object database, which is also called object-oriented database management sys-

tem(OODBMS), is a database management system restoring information in the form of objects as used in object-oriented programming. Object databases are different from relational databases which are table-oriented. Because of tighter integration with the object-oriented language, the program is easier to maintain consistency with the same representation in both OODBMS and programming language. Although relational databases might be similar to object-oriented databases, they are actually different. The object-oriented database supports objects, classes, and inheritance in the database schema and query language. There are many advantages for OODBMS compared to the relational database management system (RDBMS) such as the performance, flexibility, and development cost. And OODBMS also have some disadvantages, they have mentioned 3 disadvantages for OODBMS. First, because the usage is forced to be similar to an object-oriented language. This makes maintaining and evolving is difficult. Second, the technique for store complex type of information takes additional computational resources. Third, the absence of a standard data model leads to design errors and inconsistencies.

2. **Relational database** A relational database is the most popular database used in the world. Each row in a table has its own unique key. They can organize data into one or more tables of columns and rows, with the key identifying each row. Rows are also called records or tuples. Generally, each table represents one "entity type" (such as customer or product). The rows represent instances of that type of entity (such as "Lee" or "iPhone 6") and the columns representing values attributed to that instance (such as address or price).

   Considering the method of the organization of data, the relational database is much easier to understand and is flexible to manipulate the data. Besides SQL is easy in the relational database approach. For data organized in other structure, the query language either becomes complex or extremely limited in its capabilities. However, once the attributes of data become more and more, you'll need a large amount of tables to store your information. Therefore, the performance of relational database will decrease obviously.

3. **Graph database** A graphical database uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. Most of them are NoSQL in nature and store data in a key-value store or a document-oriented database. Graph databases are powerful tools for graph-like queries, for example, computing the shortest path between two nodes in the graph.

   As we know, relational databases are the most popular databases in the world. Compared to them, Graph databases have several advantages. A graph database is often faster for associative data set and map more directly to the structure of object-oriented applications. They can scale more naturally to large data sets as they do not typically require expensive join operations. As they less depend on a rigid schema, they are more suitable to manage ad hoc and changing data with an evolving schema.

   On the other hand, graph database also comes with some disadvantages. For example, the relational database is typically faster at performing the same operation on large numbers of data elements than graph databases.

4. **Summary** To sum up, there are several methods to store data according to the database structures. Two main directions are storing inside the database and storing out of the database. The comparison of three databases is shown in Figure 9.

   We suggest not to store binary data in the database if it is large. It may cause significant performance decrease and additional storage space. In contrast, we suggest to store binary data in the file system and record the path in the database. It may not cause the disadvantages above when large binary data store into the database, but the binary data can not automatically distribute with the database. Due to the PDF file will cost some performance issues even though it is small in size and our system has no requirement for automatic distribution. We suggest sorting the PDF file in the file system.

3.1.2 SQL and NoSQL

Every website is full of data, such as Facebook, Bank of Taiwan and official web page of National

Cheng Kung University (NCKU). The database is coming in many forms, including Object-oriented, graph and relational, which are listed above. Most of the databases come with querying languages interact with databases. SQL (Structured Query Language) is the most popular among them. It is also an American National Standards Institute (ANSI) standard. SQL is a kind of simple language. It is like English to help uo to "communicate" with database server. Therefore, even the people who are not good at programming can writing it easily.

SQL has been a single standard to support all kinds of databases for several decades. It seems good enough to let us don't need any alternatives. However, it is going to be changed. NoSQL is going to be an alternative, which means "non SQL" or "non relational". It is different from relational database management system (RDMS) in some ways. For example, NoSQL use the concept of JSON-like (JavaScript Object Notation) or name-value to store data, instead of using tables like SQL. This study already lists some differences between SQL and NoSQL shown in Figure 10.

However, SQL and NoSQL has their own advantages and disadvantages. Therefore, it should be chosen depending on data characteristic. SQL is the ideal language when projects require logical related discrete data that can be identified and data integrity is essential. On the other hand, NoSQL can be considered as an ideal language if projects require unrelated, indeterminate or evolving data and simultaneously it needs speed and scalability.

3.1.3 PostgreSQL

PostgreSQL is an open source object-relational database management system (ORDBMS). It's developed at the University of California, Berkeley. The following operating systems such as Linux, Windows and MacOS are able to install PostgreSQL. There are so many graphical user interface (GUI) like Pgadmin and PhpPgadmin we can choose. It stores most SQL:2008 data types, such as INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL or TIMESTAMP and even binary large objects including pictures, sounds, or video. It has programming language interfaces for C/C++, .NET, Java, Python, php, among others.

The following is advantages of PostgreSQL. It's free and powerful RDBMS. To deal with amounts of data , PostgreSQL is supported with open-source third-party tools compatibility extensions for designing, managing, and applying the DBMS. It has a strong and experienced community accessed through knowledge-bases sites any time for free due to the long history of PostgreSQL. PostgreSQL not only is a RDBMS system but also have features of an OODBMS. However, PostgreSQL is too simple to appear less performance than MySQL. Because of a lack of popularity, it is harder to obtain hosts or service providers that provide managed PostgreSQL examples.

3.1.4 Web Crawler

1. **Introduce to web crawler** The web crawler is a program that can automatically browse through web pages, find out the information we assigned and store them. It has ability to process the data quickly and accurate to update a very large amount of data which are constantly being updated according to Liu et al. (2012). It starts with a list of URL to visit, called the seeds. As crawler visits these URL, it identifies all the informations that we want, such as hyper links in the page and adds them to the list of URL to visit, called the crawl frontier. URL from the frontier is recursively visited according to a set of policies. If the crawler is performing archiving of websites, it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as they were on the live web, but are preserved as 'snapshots' from Du et al. (2013). We need to build up a web crawler to automatically visit a list of web page. Then find out which link in the page is valuable to download into our database.

2. **Way to create a web crawler** To create a web crawler, first we need to know how the web page works. There are two kind of web pages: dynamic page and static page. With static pages, the html of the page is directly loaded when one enters the page. With dynamic pages, the original html of the page need to be rendered by javascripts to create new html file in order to show the complete page. These two kind of page seems identical to the users, but were totally different for the crawlers. It's easy for the web crawler to find informations in static webpages. When it comes to the dynamic pages. Web crawler needs to first pass

the original html to a javascript rendering program or a light weight browser. In order to get the final html to extract informations. After we get the html files. We need to use packages such as regular expression operations to find specific information inside the file. In this case we need to find the href tags with *.pdf in it. After finding those links we need to download them one by one into the server.

3.2 Method

Author:Bernie Huan, Jim Lan, Hoang Tan, Kenny Hsu, Rahul Aditya, Tan Phat, Wei.

As mentioned in the background, our team used natural language processing in order to generate metadata. The scope of metadata processing is about producing at least three types of metadata including author name, title and abstract. This is the least we want to do within the time frame of this course. Once accomplishing the 3 types of metadata, we will extend the scope of our project, intimidate the scheme to produce other remaining metadata such as doix number, journal name, volume number and so on. If time is still on our side, we will even extend the scope of our project further and join other teams to produce a search engine. Following discussion are our plan to complete extracting the 4 first types of metadata from PDF files.

3.2.1 Title extraction

We chose python to be the program to catch the title and other metadata. First step, we import some necessary packages and functions as following:

1. re: Regular expression package is a useful package which could be applied to the string comparison.

2. os: This function can connect the python to operating system, so that we can call the path of the files and folders.

3. nltk: A natural language toolkit. We use the corpus plaintext function to build the txt file in a folder as corpus.

4. string:The string package includes a lot of classes such as lowercase, uppercase,punctuation,digits or whitespace...etc.

To begin, all PDF files are converted into txt file with which we want to deal with. The path is set and the files are stored in a specific folder. Then, regular expression in python can help capturing the first sentence in the txt files, which have been converted and stored in specific folder in previous step. The process of extracting title from the text is mainly regulated by Union_extract_title.py. The coding scheme has following steps:Importation

Database structure modes	Advantages	Disadvantages
Object-oriented database	1) Integration with object-oriented language → lower development cost good flexibility	1) Additional computational resources is needed if numbers of data is huge 2) Maintaining and evolving is difficult 3) Lack of a standard data model leads to design errors and inconsistencies
Relational database	1) Easy to learn 2) Many sources 3) Support SQL	1) Performance decreases if we have a lot of data
Graphic database (Comparison with relational database)	1) Faster in associative data sets 2) It can scale more naturally to large data sets as they do not typically require expensive join operations 3) More suitable to manage ad hoc and changing data with evolving schemas	1) Performance is worse than the relational database if numbers of data is huge

Figure 9: Advantages and disadvantages between three kind of databases.

of necessary components, setting the path, creating the directory we will write the .txt files to after stripping text, using basic command to extract titles. The results are tested and listed. We successfully produce the title of the article in the txt file.

3.2.2 Author extraction

In this part, we use the existing python packages to achieve our goal. The key point in this part is natural language processing, which we have introduced in the basic section. Also we combine the key word comparison technique to finish this task. The following step is below:

1. Section Separation: As we separate the abstract part from the full text, we extract the text above the abstract section and rename as top section. This section contains title, author and email ,etc.
2. Tokenization: After accessing the top section, we separate the text into each single word. These single words composed with a string list.
3. Part Of Speech Tag: We use the existing corpus to tag the part of speech to the words. This step is necessary since that the chunker needs the every word's part-of-speech and then chunck.
4. Chunker: In this part, we mainly chunk the noun because a noun often represents some meaningful features such as location or author.
5. Label: We check every phrase with the corpus which contain specific words like people name. Then we label the words and store within the new list. Eventually, we write the information within the xml file.

The above steps is also available to extracting the location and organization,etc when needed. The file mainly functioning for this task is Extract_Authors.py. The coding scheme for this task is quite similar to title extraction. The main different is a corpus is introduced. This scheme also include a natural language toolkit in order to extract people's name. As we test the result, our program has been able to extract authors' name.

3.2.3 Abstract extraction

Developing from previous theory mentioned in the background section, we use the python to catch the abstract. The following step is below. The pdf is converted into txt file. Thus, it will create the txt file. The work is done by hand coding. For detailed coding scheme, we have presented it in the appendix at the end of this report.

Features	SQL	NoSQL
Documents	Tables	JSON-like, name-value documents
Schema	Tables need to be defined	No specifying
Deal with data	Normalization	Denormalized
JOINS	Yes	Not require
Data Integrity	Yes	Not available
Transactions	Two or more updated	A single document
Scaling	Tricky	Easy
Querying	A declarative language	JSON data objects
Other features	i. Provide plenty of support ii. Expertise tools	i. Newer ii. Exciting technology

JSON (JavaScript Object Notation)

Figure 10: The differences between SQL and NoSQL.

After being able to read the txt file on every line, the python will detect the content of the abstract. In order to do so, we strip the text between "abstract" label and "introduction" label.

Abstract-database:including the condition

1. Capital "ABSTRACT".
2. Lower case "abstract".
3. In the sentence "Abstract—Word sense ...".
4. And so on...

In the process building "abstract database", we are aware of different papers may have different form of structures and writing style, we search through numbers of articles from different journals. Number of article found is 100, from 20 different journals including The Lancet, Progress in Energy and Combustion Science, Chemical Analysis and so. Search results for sections such as Theoretical, Methods, Results, Discussion, Conclusion, Acknowledgments are also obtained. Table below are the results of the search.

Then our program will read the txt file on every line. If the python detects the abstract-database-stop's words , it will stop to catch the sentences. abstract-database-stop:including the following conditions

1. The blank line.
2. Specific words in the beginning "Keywords".
3. And so on.

The intended output are sentences extracted to the txt file. To further validate if the program can run precisely, members randomly search for articles (10 articles) to test the programs. The program was run successfully and all abstracts was extracted from the text.

3.2.4 References URL extraction

References URL extraction is very easy and straightforward. Just use a powerful package called PDfx to complete this task. Find the package online and use it. The procedures are below:

1. First of all, a PDFx package is imported to python.
2. Second, The function in PDFx package is utilized to get the metadata and the URL of references by inputting a URL of a PDF or PDF's file name.
3. Finally, output and list the results in txt file and choose the specific directory to store it.

This method can also detects URL,arxiv and doi references. Plus,by using this method, we also can easily extract the title, page number and creation date. However,only the page number is always right in every case. The others are not totally correct. Basically, over 50 percent is not correct. Thus, among these metadata,this method is more suitable for page number.

3.2.5 Search sentences

The search sentence part is related to the search engine. After extracting the necessary information, we have to find a way to search these data. The procefure is following:

- First, The PDF is converted to the txt file.(Make the program to read the file name much easier)
- Second, read the txt files by lines.
- Third, the array is created to divide the section of the articles.
- Fourth, The array is expanded to divide the section clearly.
- Fifth, Users search the sentences, the program search all the sections by this sentence.
- Sixth, show the results.

3.3 API

We also attempt to produce an semi-automatic interface that help users to inquire certain information from our database. Currently it allows user to search for abstracts, introduction, method, result, discussion, and reference. We will continue develop the interface in the remaining week. The objectives are either covering more functions or make the results pages has more professional appearance. However, that said when we still have enough time.

4 Results and discussion

5 Conclusions and future work

References

Agerri, R., Artola, X., Beloki, Z., Rigau, G. and Soroa, A. (2014), 'Big data for Natural Language Processing: A streaming approach', *Knowledge-Based Systems* **79**, 36–42.

Alipour Hafezi, M., Horri, A., Shiri, A. and Ghaebi, A. (2013), 'Digital library interoperability: Proposing a model', *International Journal of Information Science and Management* **11**(1), 57–75.

Blascheck, T., John, M., Kurzhals, K., Koch, S. and Ertl, T. (2016), 'VA2: A Visual Analytics Approach for // Evaluating Visual Analytics Applications', *IEEE Transactions on Visualization and Computer Graphics* **22**(1), 61–70.

Cheslow, S. (2014), 'METS for the cultural heritage community: A literature review', *Library Philosophy and Practice* **2014**(1).

Creţulescu, R., David, A., Morariu, D. and Vinţan, L. (2014), Part of speech tagging with na x00ef;ve bayes methods, *in* 'System Theory, Control and Computing (ICSTCC), 2014 18th International Conference', pp. 446–451.

Dappert, A. and Enders, M. (2008), 'Using METS, PREMIS and MODS for archiving eJournals', *D-Lib Magazine* **14**(9-10).

Dave, K., Lawrence, S. and Pennock, D. M. (2003), Mining the peanut gallery, *in* 'Proceedings of the twelfth international conference on World Wide Web - WWW '03', ACM, p. 519.

Dempsey, L. and Computer, O. (2015), 'Specification for resource description methods , Part 1 . A review of metadata : a survey of current resource description formats', (October).

Du, Y., Pen, Q. and Gao, Z. (2013), 'A topic-specific crawling strategy based on semantics similarity', *Data and Knowledge Engineering* **88**, 75–93.

Duval, E., Hodgins, W., Sutton, S. and Weibel, S. L. (2002), 'Metadata principles and practicalities', *D-Lib Magazine* **8**(4), 16.

f. Wang, Y., j. Zhang, Y., t. Xu, Z. and Zhang, T. (2006), Research on dual pattern of unsupervised and supervised word sense disambiguation, *in* '2006 International Conference on Machine Learning and Cybernetics', pp. 2665–2669.

Gartner, R. (2013), 'Parliamentary Metadata Language: An XML Approach to Integrated Metadata for Legislative Proceedings', *Journal of Library Metadata* **13**(1), 17–35.

Grehan, M. (2002), 'How Search Engines Work', *Search Engine Marketing: The Essential Best Practice Guide* p. 57.

Guenther, R. and McCallum, S. (2003), 'New Metadata Standards for Digital Resources: MODS and METS', *Bulletin of the American Society for Information Science and Technology* **29**(2), 12–15.

Guo, Y., Che, W., Liu, T. and Li, S. (2010), Semi-supervised domain adaptation for wsd: Using a word-by-word model selection approach, *in* 'Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on', pp. 680–687.

Kules, B., Kustanowitz, J. and Shneiderman, B. (2006), 'Categorizing Web Search Results into Meaningful and Stable Categories Using Fast-Feature Techniques', *Digital Libraries, Joint Conference on* **0**, 210–219.

Liu, J. N. K., Choi, K. C. and Chai, J. Y. (2012), 'Development of an intelligent distributed news retrieval system', *International Journal of Knowledge-Based and Intelligent Engineering Systems* **16**(2), 129–140.

(NISO), N. I. S. O. (2012), 'The Dublin Core metadata element set', **Version 1.**

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D. and Ferro, L. (2003), The TIMEBANK Corpus, *in* 'Corpus linguistics', Vol. 2003, p. 40.

Ramos-Soto, A., Bugarín, A. and Barro, S. (2015), 'On the role of linguistic descriptions of data in the building of natural language generation systems', *Fuzzy Sets and Systems* **285**, 1–21.

Sak, H., Saraçlar, M. and Güngör, T. (2010), Morphology-based and sub-word language modeling for turkish speech recognition, *in* '2010 IEEE International Conference on Acoustics, Speech and Signal Processing', pp. 5402–5405.

Saur??, R. and Pustejovsky, J. (2009), 'Factbank: A corpus annotated with event factuality', *Language Resources and Evaluation* **43**(3), 227–268.

Schultze, U. (2000), 'A Confessional Account of an Ethnography About Knowledge Work', *MIS Quarterly* **24**(1), 3–41.

Tague, J., Beheshti, J. and Rees-Potter, L. (1981), 'The Law of Exponential Growth: Evidence, Implications and Forecasts', *Library Trends* **30**(1), 125–149.

Tamine, L., Jabeur, A. and Bahsoun, W. (2009), 'Flexible Query Answering Systems', *Flexible Query Answering Systems* **5822**(JANUARY), 88–98.

Tun Thura Thet, Na, J.-C. and Khoo, C. S. G. (2010), 'Aspect-based sentiment analysis of movie reviews on discussion boards', *Journal of Information Science* **36**(6), 823–848.

Underwood, J. and Watson, A. (2003), 'An XML metadata approach to seamless project information exchange between heterogeneous platforms', *Engineering, Construction and Architectural Management* **10**(2), 128–145.

Wiebe, J., Wilson, T. and Cardie, C. (2005), 'Annotating expressions of opinions and emotions in language', *Language Resources and Evaluation* **39**(2-3), 165–210.