Executive Summary

Aim and Objectives:

As per the requirements of our client (Lnx), this prototype system is expected to extract abstract from a medicine conference website and complete the author-organization relationship disambiguation.

Type of Project:

This project is 70% practical and 30% research. The research part is focus on the algorithm of author-organization relationship disambiguation. Disambiguation is not a new topic, but all the research is focus on entity disambiguation, not on the relationship disambiguation. The practical part is about software design, coding and testing.

Implementation Summary:

I have implemented the core algorithm based on previous research, the abstract extraction module, the xml parse and generating module, the database design, the database operation by JDBC, the web module to send a request and bring back its content, the Bing search module which use Bing search API directly.

High-light Features or Performance:

In this project, I have done a lot. Among all the efforts, there are 5 features I am really proud of.

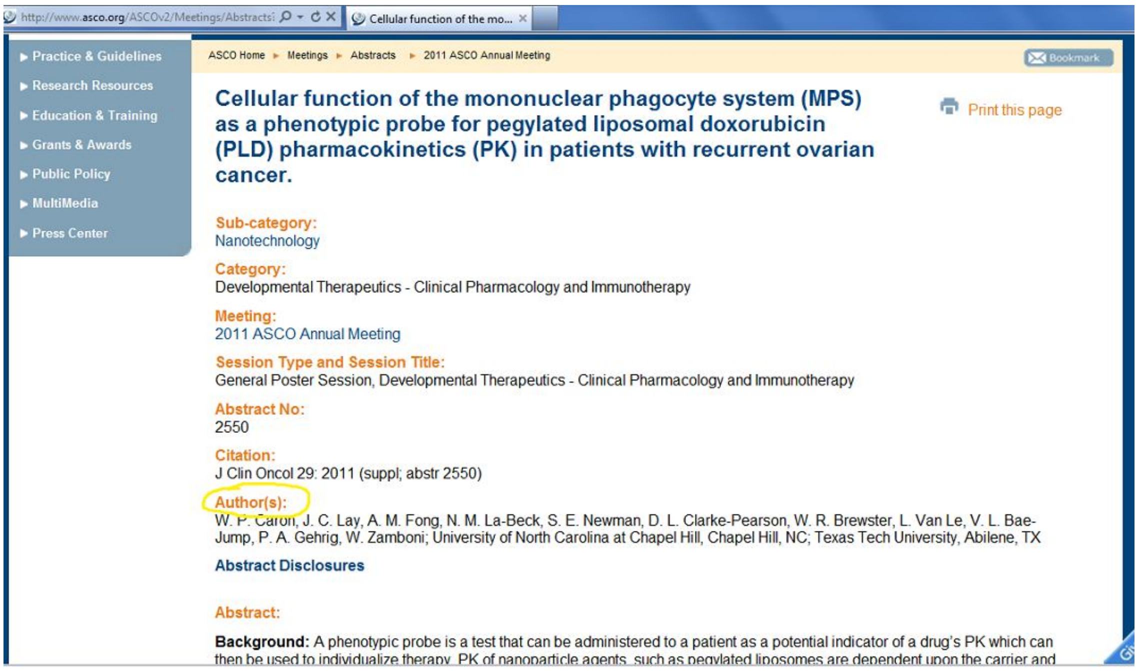
1. Design the author-disambiguation algorithm and it works well.
2. Design the feature ‘Continue from Lost’, which make the system continue from where it was end last time. This program needs to access web page frequently but the network will be disconnected suddenly. With this feature, no matter what causes the system to quit, it can continue from nearby.
3. Abstract extract module is perfect. It is totally automated, correct and fast. Moreover, it is compatible with different format.
4. Smart design makes this system flexible and efficient. As a prototype system, I not only implement the core algorithm, but consider the future extension. Configuration file is used to set some system parameters which have big impact on system’s performance. Bing search module is implemented for the future need.
5. Suppose several feasible enhancements for the future extension.

Table of Contents

1. Introduction

Author disambiguation has been a growing concern in bibliographic areas for several years. Many researches and experiments have been done to disambiguate the authors, but little has been done on the disambiguation of author-organization relationship. A review of recent literature reveals that most of the author disambiguation researches are designed for citation. Therefore organization information is usually ignored. Due to the market requirement, we designed a practical system with an experimental algorithm to implement the author-organization disambiguation. In this dissertation, we will discuss the system's structure, the core algorithm and the evaluation.

For the research company like Lnx which is trying to build a bibliographical network, a clear matching of author-organization relationship is important. However, the mapping is unclear sometimes. It sounds unreal, but it is true unfortunately. Our client (Lnx) used to get information from ASCO (American Society of Clinical Oncology) international medical conference, which only lists all the participants and their organizations but does not provide a one-to-one mapping between the authors and their organizations, as shown in Figure 1-1 below.



Cellular function of the mononuclear phagocyte system (MPS) as a phenotypic probe for pegylated liposomal doxorubicin (PLD) pharmacokinetics (PK) in patients with recurrent ovarian cancer.

Sub-category:
Nanotechnology

Category:
Developmental Therapeutics - Clinical Pharmacology and Immunotherapy

Meeting:
2011 ASCO Annual Meeting

Session Type and Session Title:
General Poster Session, Developmental Therapeutics - Clinical Pharmacology and Immunotherapy

Abstract No:
2550

Citation:
J Clin Oncol 29: 2011 (suppl; abstr 2550)

Author(s):
W. P. Caron, J. C. Lay, A. M. Fong, N. M. La-Beck, S. E. Newman, D. L. Clarke-Pearson, W. R. Brewster, L. Van Le, V. L. Bae-Jump, P. A. Gehrig, W. Zamboni; University of North Carolina at Chapel Hill, Chapel Hill, NC; Texas Tech University, Abilene, TX

Abstract Disclosures

Abstract:

Background: A phenotypic probe is a test that can be administered to a patient as a potential indicator of a drug's PK which can then be used to individualize therapy. PK of nanoparticle agents, such as pegylated liposomes are dependent upon the carrier and

Figure 1-1: Example of unclear author-organization mapping

In the "Author(s)" section of figure 1-1, there are in total 11 authors but only 2 organizations. Apparently there is no one-to-one mapping. Even the number of author is equal to the number of organization, we can hardly say that they are one-to-one mapped well because we do not know how these names are ordered. The organizations might be in the same order as the authors, but they might also be displayed in an alphabetic order or totally randomly. Based on the manually research result shown in Table 1-1, organizations of the abstracts in ASCO website are ordered randomly.

Author	Title	Organization
W. Zamboni	Cellular function of the mononuclear phagocyte system (MPS) as a phenotypic probe for pegylated liposomal doxorubicin (PLD) pharmacokinetics (PK) in patients with recurrent ovarian cancer	University of North Carolina at Chapel Hill, Chapel Hill, NC

Table 1-1: Fact data for author W. Zamboni

If organization is in the author's order, the organization of the last author 'W. Zamboni' should be '*Texas Tech University, Abilene, TX*'. If it is in alphabetic order, '*University of North Carolina at Chapel Hill, Chapel Hill, NC*' should be after '*Texas Tech University, Abilene, TX*'. To sum up, there is no mapping between authors and organizations in ASCO's abstracts.

Based on above facts, our system resumes that the order of authors are totally irrelevant with the order of organizations. So, we design an algorithm which we call **Two Stage algorithm**: 1) the first step is to find out all the abstracts written by the same author; 2) the second step is to find the organization for each author based on the last step's data. Either step can affect the performance of the system, but the first step is the most important because the second step is built on it.

To make the system automated, another important and necessary part is the extraction module which is expected to extract the abstracts from ASCO website correctly and automatically. Because each website has its own web format for displaying the abstract, there is no way to design a module that is suitable for all sites. At current stage, the system is only able to extract information from ASCO website, as per client's requirement.

A good design is very important to a system. So before we start to implement, we spend a lot of time on the design. All the ideas are based on client's requirement. In Chapter 3, we discuss the system design, from the overall structure design to core algorithm design.

Chapter 4 shows some details of the core modules.

Chapter 5 is about testing. All the key test data can be found here. In this chapter, you can see what the performance of the system or one module is. We also analyze the results.

Chapter 6 tells how to evaluate the system and how well our system performs.

Chapter 7 is about future extension. Due to the time and effort limitation, this system works well as a prototype system, but needs some improvement if we want to make it commercial. In this chapter, enhanced algorithm and new research direction are discussed briefly.

2. Research Review

2.1Author Disambiguation

Ambiguous authors can be caused by different reasons: spelling error, orthographic variants, identical names, pseudonyms etc. But they can be classified into two categories: homonyms and synonyms [1].

Homonyms can happen when several persons having the identical name label, this normally happens because:

- Different people have the same full name. Searching "David R. Wood" in Google Scholar, we can get pager "Chronic Fatigue in Primary Care" and paper "An algorithm for finding a maximum clique in a graph". In fact, the two David are different people, because one David is working in Brooke Army Medical Centre and the other is in Monash University, Australia.

- Different people share the same abbreviation. "David R. Wood" mentioned above is abbreviated as "DR Wood", and "D. Robley Wood" for the book "The Impact of Comprehensive Planning on Financial Performance" is also abbreviated as "DR Wood"

Synonyms are a complementary category. It is about the fact that an individual may have several name-labels. In the real world, several reasons can cause it:

- One may changes her last name after marriage or due to religious reason. For example, professor Claudia Popien in Aachen University of Technology used her full name before, then changed to "Claudia Linnhoff-Popien" after her marriage.

- One may have pen-names.

- One's name may be different due to translation reasons, especially Asian names. For example, "Liu" is a common family name in China, but it is usually spelled as "Lyu" for the convenience of pronunciation.

- One can be represented by different names due to spelling error or different spelling habits.

In general, if ambiguous name exists somewhere, both homonyms and synonyms exist. They are tightly connected but quite different, so researchers usually address them separately. Our system targets the synonyms only.

2.2Related Work

Research has been widely performed in database, information retrieval and digital library area. Reviewing these work is very important, because it enables us to know what they tried to solve, as well as how and to which extent they solved. Eventually it will help us to design our system and to improve it.

2.2.1 Database Area

In database area, name disambiguation is concerned with the duplicated record in the same database or from different data source.

Record linkage theory was formalized by Fellegi and Sunter in statistics [2], soon the entire theory was transferred to name disambiguation, mainly used in big databases.

Assume we have two databases A and B, and some data are in fact referring to the same entity. We also define two data sets U and M. M is to store those matching record, and U is for others. The task of record linkage is to assign each record in A and B into data set U or M [3]. When it is used for name disambiguation, a simple but direct example will be like this: an entry 'D. Smith' is both A and B, we have to find out if the two "D. Smith" are the same people.

To achieve this, record linkage compares the attributes of each entry to see how similar they are. In this case, the attributes are the other metadata in the database for this entry, which may probably the email address, home address, phone number etc. Since these metadata are generally stable, so record linkage can achieve a higher accuracy.

Duplicate Record Detection is another common approach in database. It aims to identify and eliminate duplicate records in large databases, it concentrates on the detection of duplicated records, especially on those whose values are difficult to compare.

If two records are in different databases, they are difficult to compare when:

1) They have different schema. For example, address "44 West Fourth Street" [4] may be saved in one database as a single field "address", while in another database it may save as multi-fields, like "house number ", "street name".

2) They have different data representation, even their schemas are the same. For example, the address in 1) can be "44 West Forth Street" in one database, and "W. 4th Street, 44" in the other database.

2.2.2 Information Retrieval Area

Ontology [5] was originally in philosophical discipline. But in the recent decades, it was introduced into computer discipline. Gruber described it as a "systematic account of existence" [5], and he proposed that Ontology should model a set of concepts and the relationship between these concepts within a certain domain. Firstly, it builds a concept model base on given data; Then it explicit the constraints or relationship between these concepts; Then all the information is formalized so that it can be finally shared by the whole domain.

Cross-document coreference [6] can be caused by ambiguous name in a document or cross the document. It uses clustering method to do the disambiguation. The author of [7] used SVM method to solve this problem. If there are two entries from two documents for a same name, the author builds word vector from the sentence where the name is mentioned, then compare to see if they are referring to the same person.

2.2.3 Digital Library Area

Heuristics Graph was proposed in [8]. It is designed for a specified situation of synonyms: one person has different first name labels.

Firstly, they normalized the name to be a standard format: Initial character of first name + the full family name. i.e., "David Smith" will be normalized as "D Smith". Then they perform the mating algorithm, a simple but effective regular express compare. Next they build a graph, each node in this graph is a name label, and the link between two nodes has a weight representing how many names they match with. At the end, group the nodes with the heaviest weight.

They succeed in reducing the number of distinct authors “from 3076 to 1007” [8] in ACM data. Considering they only use the names themselves without any other information, it is really a great achievement.

Mixed Citation is caused by homonyms. In digital libraries, if different authors have the same name labels, their citation may be mixed together.

Baseline algorithm was used in [9] to solve this problem. Given an author set A and a citation set C. For each citation c in C, perform the similarity measure between c and each author, the author with the largest similarity is treated as the author of citation c. It is intuitive but not effective. To improve the efficiency, a sampling-based join approximation method which was developed in [10] is used to reduce the size of initial data set.

Split Citation is caused by synonyms. If one author has several name labels, his citations may be split into different name categories.

In paper [9], the author uses co-author information to check whether two names are referring to the same people. For name A and name B, perform the distance measure for A’s co-author list and B’s co-author list. If the value is smaller than a pre-defined threshold, A is treated as B’s variant.

Social Network-based name disambiguation is proposed in [11] for the synonym detection. The author builds a co-author graph. Each node in the graph is one author and the line between two authors means they have published together. Based on a serial of assumption, for example, if author A published with Q, author B also published with Q, then it is believed that A and B have some kind of relationship, the approach can successfully solve this synonym problem.

2.3Summary

Different approaches were designed to solve author disambiguation problem, but none of them tried to disambiguate the relationship. By reviewing the prior research, we find out co-author information is common used. Our project aims to finish the author-organization relationship, so we proposed a new method of author disambiguation. Our contribution is the practical application of a novel author-organization relationship disambiguation solution.

3. System Design

In the real word, names are normally used to distinguish people. When two persons have the identical names, other people tend to add other information like occupation, age or something else to make them different: Professor James vs. Doctor James; Old James vs. Young James. The same things happen in the publications — when two abstracts have the same author name, we have to use other information like the organization, title, research keywords etc, to check whether they are actually written by the same author. Recent research shows co-author information is commonly used. We decide to use co-author information too, but in a quite different way. After all, relationship disambiguation is our project's final object. We designed what we called Two Stage algorithm.

As per the customer's requirements, the dataset of our system is an international medicine conference's website. Unlike most applications whose dataset is database, our system has to deal with the web pages. This makes our system more complicated.

Designing a system is not easy because we have too many options and each option has its advantages and disadvantages. For example, If we want to save the result, we can store the information into a plain text file, an excel file, database or other format. All these ways are able to work as we expect, but each one is the best for the system? In this chapter, we will outline our system and discuss a little bit about our choice.

3.1Algorithm Design

To complete the author-organization disambiguation, a straight-forward way is to search on the author's name then find out his organization. In our system, we use the same idea and call it Two Stage algorithm. Two Stage algorithm splits the disambiguation process into two stages:

1) Stage 1 is to execute the author disambiguation. By doing this, we should get all the abstracts written by that author.
2) Stage 2 is to confirm the organization for that author according to the data obtained in stage 1. After this stage, we can get a clear author-organization relationship: one-to-one mapped or not found.

In the following sections, we will introduce these two steps.

Note:

For convenience, the abstract we are trying to do the relationship disambiguation is called 'sample abstract'. The abstract which has already finished with step 1 is called 'clustered abstract'. In other words, sample abstract and clustered abstract are written by the same author.

3.1.1 Author Disambiguation

In stage 1, we need to do the author disambiguation. Two authors may have the same name identity, but statistical data indicates they have less possibility to publish with the same co-authors. Based on this fact, we use co-authors to do the author disambiguation.

We define two kinds of co-authors relationship: direct co-author relationship and indirect co-author relationship. To get the good understanding of the definitions, we can take use of graph to show the relationship intuitively.

Each node of a graph represents one author, and the edge between two nodes means they belong to the same paper. Figure 3-1 demonstrates how an abstract is converted into a graph, which looks a little complicated if all the co-author relationships get displayed. If we are only interested in some of the authors, we can ignore the relationship among other authors. For example, when we focus on author "Shen", all the dashed edges can be omitted.

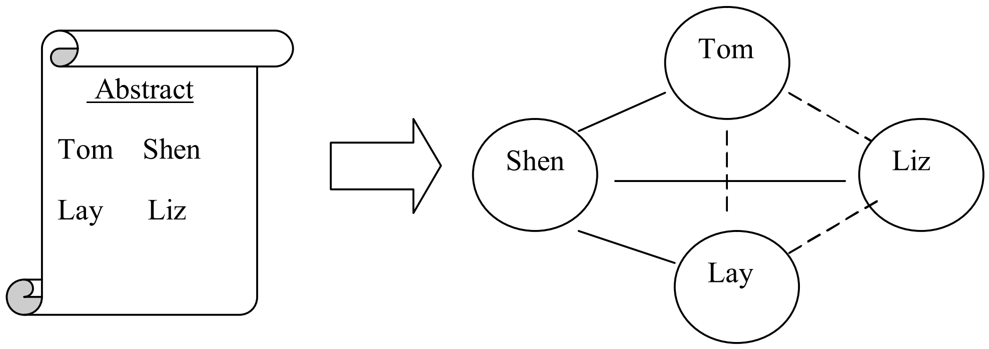


Figure 3-1:Left Abstract Represent by Right Graph

- **Direct co-author relationship:** If two authors publish with the same other person, we say these two authors have direct co-author relationship. For example, if author A writes paper P_1 with author X and another author B writes paper P_2 with author X, author A and author B then have the direct co-author relationship. The relationship graph is as Figure 3-2.

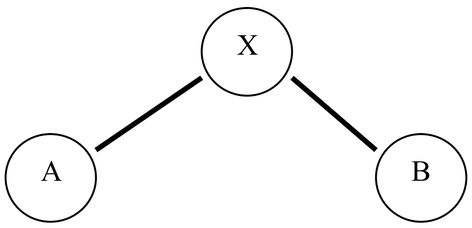


Figure 3-2: A and B have direct co-author relationship

- **Indirect co-author relationship:** Given two authors, if one author's co-author publishes with one of the other author's co-authors, we say these two authors have indirect co-author relationship. For example, paper P_1 has author A and other authors $A_1, A_2…A_m$; and paper P2 has author B and other authors $B_1, B_2…B_n$. If any Ai among { $A_1, A_2…A_m$} publishes with any B_j among { $B_1, B_2…B_n$}, author A and author B have indirect co-author relationship. The relationship graph is as Figure 3-3.

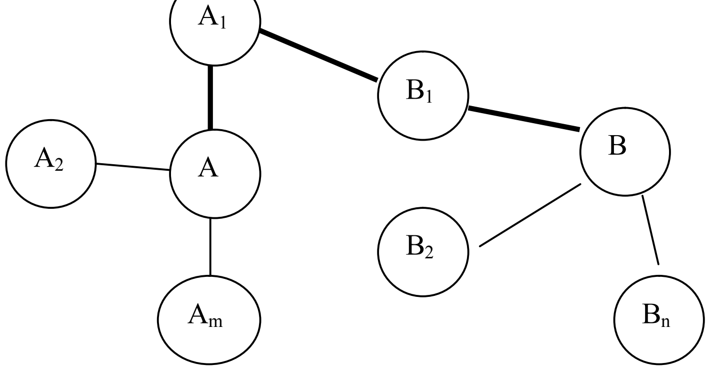


Figure 3-3: A and B have indirect co-author relationship.

Bold lines shows how they are connected

For the authors with the same names, we treat them as identical people if and only if they have direct or indirect co-author relationship.

3.1.2 Organization Confirmation

A name is an identifier. Just as authors can change their names, so can the organizations. An organization can change its name due to different reasons. The common reasons include:

1) The adjustment of management. For example, "Johns Hopkins University" changed its name to "John Hopkins University".
2) The change of research field. For example, American Board of Podiatric Surgery (ABPS) changed its name to American Board of Foot and Ankle Surgery (ABFAS) because the latter reflected the organization's research advantages.

If one author happens to publish two papers, one is before the name change and the other is after the name change. It is difficult for our system to get the correct conclusion unless the name change information has already been predefined. Generally speaking, the name of organization is quite stable, so our system will not take the name change of organization into account.

After stage 1, we will get a collection of that author's clustered abstracts, which should contain the organization information in most situations. In the abstract, the organization exists in two statuses: it is connected with the author or it is not.

In sample abstract, we have several authors and several organizations, but they do not match with each other. If the relationship is clear in clustered abstract, we can get the organization and compare it to each organization of sample abstract. If we find one identical organization, this organization is what we are looking for, because Two Stage algorithm guarantees that they are written by the same author,

If the relationship in clustered abstract is not clear, we can apply intersection operation on the author set and organization set. If there are more than one clustered abstract, we can keep intersection until no clustered abstract is left or we got only one author and one organization – the organization is what we are looking for.

Another thing we have to consider is the organization compare. A straight string compare is the simplest way, but it requires two strings have the same character sequence. In the real world, an organization may be displayed in different format. For example, "Health Sciences Institute" may be written as "The Institute of Health Sciences"; "Department" is commonly replaced by its abbreviation "Dept.". The straight string compare will fail in this case.

One alternative way is to break on organization into words. Then check if each word exists in another organization. If every single word is contained in another organization, we confirm the two organizations are the same. This way is better than the straight string, but has its limitation. Consider two organizations "Health Institute, Bristol University, Bristol, UK" and organization "Bristol University Institute, Bristol, UK". If we use the compare method, we will think they are the same because each word in {Bristol, University, Institute, Bristol, UK} exists in organization "Health Institute, Bristol University, Bristol, UK". This incorrect compare result is generated because we just do the literature compare without considering its context.

Organization name is just a serial of meaningless words for computers, but it is meaningful for the human being. When mention something, people always have fixed habits. For example, address is ordered from the house number to city in western countries, while eastern countries arrange it from the city to house number. The main point is that address is of a certain order. So is the organization. Organization consists of parts or all of the following elements:

- Department information
- Organization information. The name of the research company or university
- City. Where the organization locates
- Country. Which country the organization belongs to

Each part is separated with a separator, which is usually "," or ";". We treat each part as a meaningful unit and compare them as a whole. For each part, we can use the word compare mentioned in last graph. If each meaningful unit is the same, we believe the two organizations are the same.

3.2Component Design

In a system, algorithm is just part of it. To meet the client's requirements and to execute the author-organization relationship disambiguation automated and successfully, our system needs to interaction with different outer components. The components' number and type is depends on the system's function. Our system considers to use the following components: the data set to get sample abstract; the file system or database to save the output and the search engine to collect information.

3.2.1 Data Set

For our system, data set is all the sample abstracts. Our client is a research company and always interested in the abstracts published in ASCO annual meeting, so we use ASCO website (http://www.asco.org/) as the data set. ASCO is a very professional and far-reaching oncology society. Each year, thousands of hundreds latest research papers are published in the meeting.

3.2.2 File Format

When it comes to the file store format, the first one that flashes into our minds probably is the plain text file. As the most universal format, plain text file can be opened, read and edited in all the platforms without any special requirements. Almost all the OS provides the free support for plain text, like notepad under Windows, edit command under DOS, vi or vim under Linux/Unix, SimpleText under Mac OS. Basically a plain text file is a file that contains characters. It does not define an unified format, so you can store the result as "author-organization", "author : organization", "author <-> organization" or other format you like, which requires extra efforts in parsing.

Except the plain text file, we have many other choices, like DOS file, HTML file, EXCEL file, Word file. They are all good supported by parts or all OS. We finally choose XML(Extensible

Markup Language) format. XML is a texture data format with an international specification, so it is platform-independent and easy to read by both the human beings and the computers.

3.2.3 DBMS

DBMS (Database Management System) is a software package with computer program that control the basic database operation and user management. Nowadays, the most common database is relation database. MS SQL and Oracal are the most widely used commercial product. We did not consider them because

1) Database operation is not complicated in our system. We use database to save middle results. The data is well structured and only basic SELECT, INSERT will be used in the system. Any DBMS can meet our requirements, so we do not need to use the complex ones.

2) They are not free to use. This project is for academic use, we do not have money to pay for any extra costs.

So at the end we decide to use MySQL. This product is a open source software and it provides as many functions as MS SQL and Oracal.

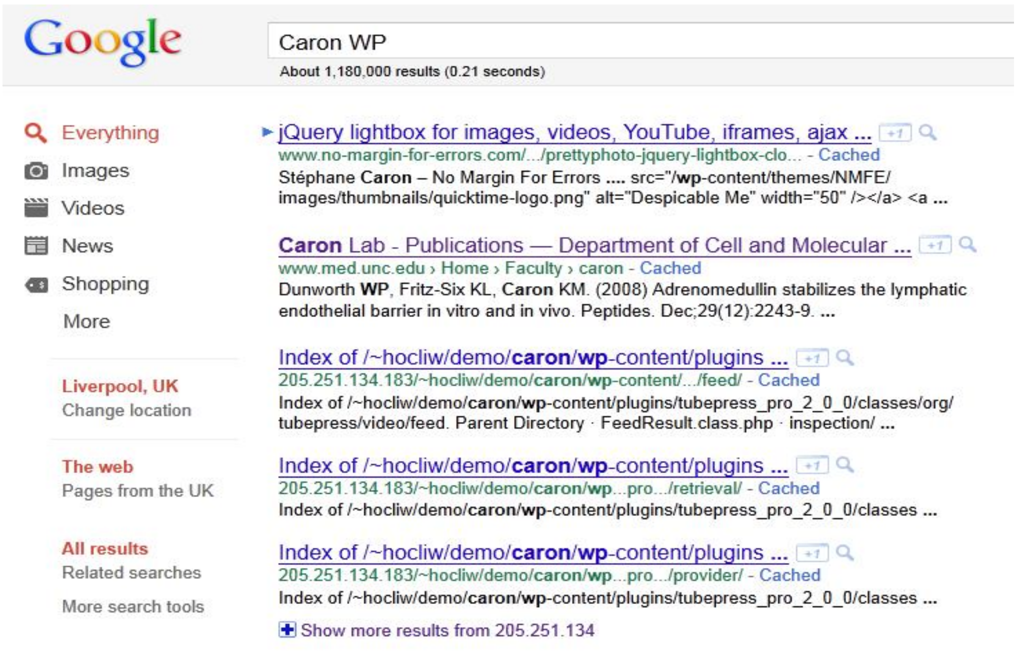
3.2.4 Search Engine

Search engine is not a new thing: even as early as in 1996, the survey indicated that among the 5 most visited websites, 3 were search engines [12]. Nowadays, with the exponential increase of the information in internet, search engine plays a more and more important role. It is well known that the information returned depends on the search engine itself and the query used.

Search engine is designed to retrieve information (text, image or other data type) from web. There are more than 30 search engines on the internet. Some are limited to question answering service, like ChaCha, Ask; while some provides overall service, like Google, Bing. Some are world-wide, while some are especially for a certain country, like Baidu in China, Daum in South Korea.

Despite the service difference, crawl-based search engine works in the same way. It uses a web crawler (also known as a spider) to get knowledge about existing web automatically, analyze them to form index structures [13]. When search keywords are provided by the user, search engine retrieves the information, rank them using their own ranking algorithm and finally display the results to the user. It implies that for the same query, the results may be different.

To check the performance of each search engine, we search an author's name 'Caron WP' in Google and Bing. Figure3-4 displays the first 5 results returned by Google, none is relevant with that author.



Google

Caron WP

About 1,180,000 results (0.21 seconds)

jQuery lightbox for images, videos, YouTube, iframes, ajax ...
www.no-margin-for-errors.com/.../prettyphoto-jquery-lightbox-clo... - Cached
Stéphane **Caron** – No Margin For Errors .... src="/**wp**-content/themes/NMFE/ images/thumbnails/quicktime-logo.png" alt="Despicable Me" width="50" /></a> <a ...

Caron Lab - Publications — Department of Cell and Molecular ...
www.med.unc.edu › Home › Faculty › caron - Cached
Dunworth **WP**, Fritz-Six KL, **Caron** KM. (2008) Adrenomedullin stabilizes the lymphatic endothelial barrier in vitro and in vivo. Peptides. Dec;29(12):2243-9. ...

Index of /~hocliw/demo/**caron**/**wp**-content/plugins ...
205.251.134.183/~hocliw/demo/**caron**/**wp**-content/.../feed/ - Cached
Index of /~hocliw/demo/**caron**/**wp**-content/plugins/tubepress_pro_2_0_0/classes/org/ tubepress/video/feed. Parent Directory · FeedResult.class.php · inspection/ ...

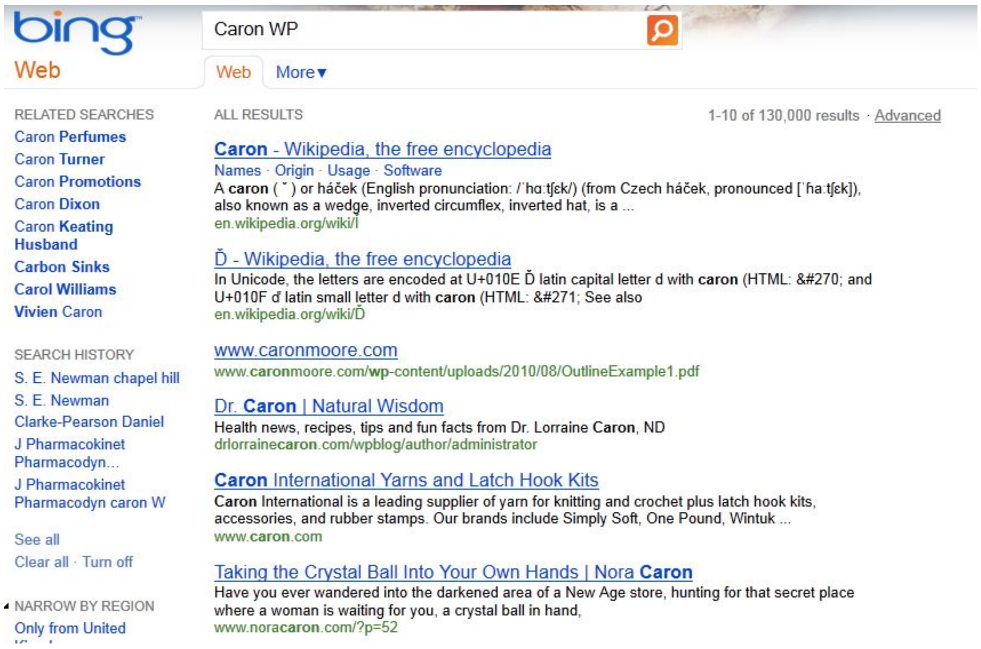
Index of /~hocliw/demo/**caron**/**wp**-content/plugins ...
205.251.134.183/~hocliw/demo/**caron**/**wp**...pro.../retrieval/ - Cached
Index of /~hocliw/demo/**caron**/**wp**-content/plugins/tubepress_pro_2_0_0/classes ...

Index of /~hocliw/demo/**caron**/**wp**-content/plugins ...
205.251.134.183/~hocliw/demo/**caron**/**wp**...pro.../provider/ - Cached
Index of /~hocliw/demo/**caron**/**wp**-content/plugins/tubepress_pro_2_0_0/classes ...

Show more results from 205.251.134

Figure 3-4: First 5 Results for author "Caron WP" in Google

Figure3-5 displays the first 5 results returned by Bing. It looks better because once of the result 'Dr. Caron' is obviously a name and might be the author we are searching for.



bing
Web

Caron WP

ALL RESULTS 1-10 of 130,000 results · Advanced

Caron - Wikipedia, the free encyclopedia
Names · Origin · Usage · Software
A **caron** (ˇ) or háček (English pronunciation: /ˈhɑːtʃɛk/) (from Czech háček, pronounced [ˈɦaːtʃɛk]), also known as a wedge, inverted circumflex, inverted hat, is a ...
en.wikipedia.org/wiki/ˇ

Ď - Wikipedia, the free encyclopedia
In Unicode, the letters are encoded at U+010E Ď latin capital letter d with **caron** (HTML: &#270; and U+010F ď latin small letter d with **caron** (HTML: &#271; See also
en.wikipedia.org/wiki/Ď

www.caronmoore.com
www.**caron**moore.com/**wp**-content/uploads/2010/08/OutlineExample1.pdf

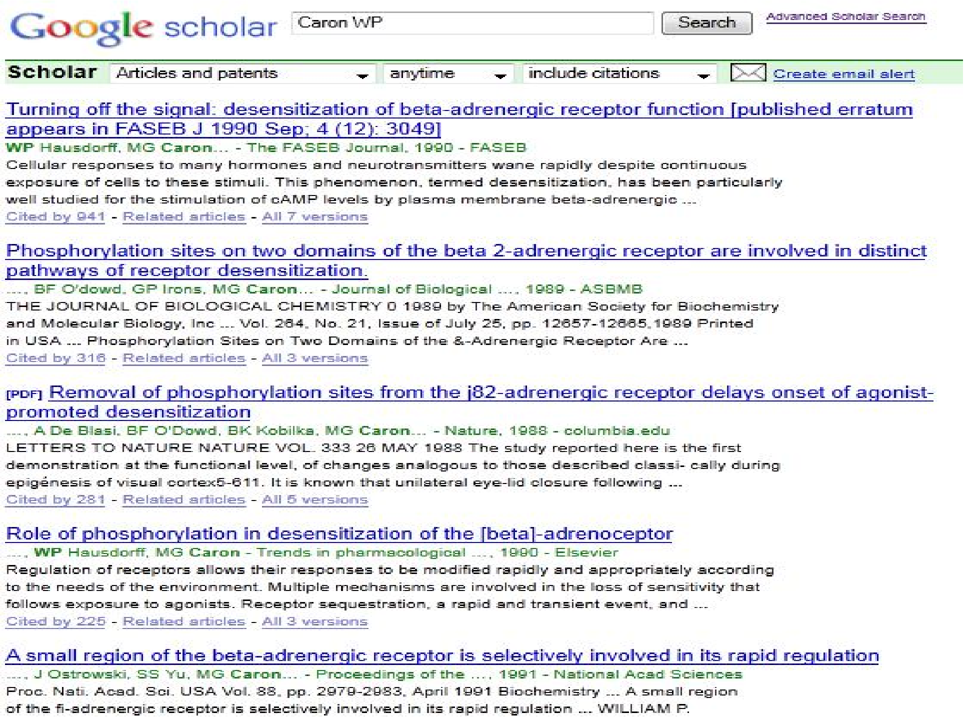
Dr. **Caron** | Natural Wisdom
Health news, recipes, tips and fun facts from Dr. Lorraine **Caron**, ND
drlorraine**caron**.com/**wp**blog/author/administrator

Caron International Yarns and Latch Hook Kits
Caron International is a leading supplier of yarn for knitting and crochet plus latch hook kits, accessories, and rubber stamps. Our brands include Simply Soft, One Pound, Wintuk ...
www.**caron**.com

Taking the Crystal Ball Into Your Own Hands | Nora **Caron**
Have you ever wandered into the darkened area of a New Age store, hunting for that secret place where a woman is waiting for you, a crystal ball in hand,
www.nora**caron**.com/?p=52

Figure 3-5: First 5 Results for author "Caron WP" in Bing

If the search engine can only search in the academic area, the result will be much more useful. We try the same search keywords in Google Scholar and Bing Academic and. All the results are academic, but not really useful. Figure 3-6 displays the first 5 results returned by Google Scholar, none of them is connected with 'Caron WP'. The same result happens to Microsoft academic whose returned information is list in Figure 3-7.



Google scholar | Caron WP | Search | Advanced Scholar Search

Scholar | Articles and patents | anytime | include citations | Create email alert

Turning off the signal: desensitization of beta-adrenergic receptor function [published erratum appears in FASEB J 1990 Sep; 4 (12): 3049]
WP Hausdorff, MG Caron... - The FASEB Journal, 1990 - FASEB
Cellular responses to many hormones and neurotransmitters wane rapidly despite continuous exposure of cells to these stimuli. This phenomenon, termed desensitization, has been particularly well studied for the stimulation of cAMP levels by plasma membrane beta-adrenergic ...
Cited by 941 - Related articles - All 7 versions

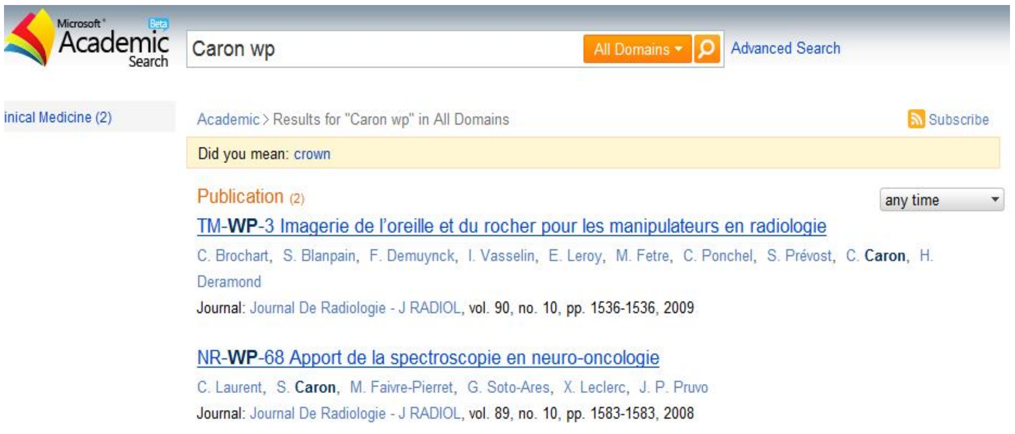
Phosphorylation sites on two domains of the beta 2-adrenergic receptor are involved in distinct pathways of receptor desensitization.
..., BF O'dowd, GP Irons, MG Caron... - Journal of Biological ..., 1989 - ASBMB
THE JOURNAL OF BIOLOGICAL CHEMISTRY 0 1989 by The American Society for Biochemistry and Molecular Biology, Inc ... Vol. 264, No. 21, Issue of July 25, pp. 12657-12665,1989 Printed in USA ... Phosphorylation Sites on Two Domains of the &-Adrenergic Receptor Are ...
Cited by 316 - Related articles - All 3 versions

[PDF] Removal of phosphorylation sites from the j82-adrenergic receptor delays onset of agonist-promoted desensitization
..., A De Blasi, BF O'Dowd, BK Kobilka, MG Caron... - Nature, 1988 - columbia.edu
LETTERS TO NATURE NATURE VOL. 333 26 MAY 1988 The study reported here is the first demonstration at the functional level, of changes analogous to those described classi- cally during epigénesis of visual cortex5-611. It is known that unilateral eye-lid closure following ...
Cited by 281 - Related articles - All 5 versions

Role of phosphorylation in desensitization of the [beta]-adrenoceptor
..., WP Hausdorff, MG Caron - Trends in pharmacological ..., 1990 - Elsevier
Regulation of receptors allows their responses to be modified rapidly and appropriately according to the needs of the environment. Multiple mechanisms are involved in the loss of sensitivity that follows exposure to agonists. Receptor sequestration, a rapid and transient event, and ...
Cited by 225 - Related articles - All 3 versions

A small region of the beta-adrenergic receptor is selectively involved in its rapid regulation
..., J Ostrowski, SS Yu, MG Caron... - Proceedings of the ..., 1991 - National Acad Sciences
Proc. Natl. Acad. Sci. USA Vol. 88, pp. 2979-2983, April 1991 Biochemistry ... A small region of the fi-adrenergic receptor is selectively involved in its rapid regulation ... WILLIAM P.

Figure 3-6: First 5 Results for "Caron WP" in Google Scholar



Microsoft Academic Search Beta | Caron wp | All Domains | Advanced Search

inical Medicine (2)

Academic > Results for "Caron wp" in All Domains | Subscribe

Did you mean: crown

Publication (2) | any time

TM-WP-3 Imagerie de l'oreille et du rocher pour les manipulateurs en radiologie
C. Brochart, S. Blanpain, F. Demuynck, I. Vasselin, E. Leroy, M. Fetre, C. Ponchel, S. Prévost, C. Caron, H. Deramond
Journal: Journal De Radiologie - J RADIOL, vol. 90, no. 10, pp. 1536-1536, 2009

NR-WP-68 Apport de la spectroscopie en neuro-oncologie
C. Laurent, S. Caron, M. Faivre-Pierret, G. Soto-Ares, X. Leclerc, J. P. Pruvo
Journal: Journal De Radiologie - J RADIOL, vol. 89, no. 10, pp. 1583-1583, 2008

Figure 3-7: First 5 Results for "Caron WP" in Microsoft Academic

About search tests reveal that Bing and Google can retrieve much more results, but how to find the information we are interested in will be more difficult. Google Scholar's results are all academic, seems suitable for our system. Microsoft Academic is about focus on academic, but Microsoft has stopped its support for years. So the returned results are usually quite old.

3.2.5 Bibliographic Database

A bibliographic database is a database of bibliographic records. Records in a bibliographic database can present articles and conference papers. They are generally designed to be retrieved by author, title or keyword. Doing a query in a bibliographic database means to search the information from a wide range of digital sources, such as journals, newspaper articles, conference proceedings, patents, books, reports etc.

Bibliographic databases are normally provided by research or education organizations, but some big publishing companies also build their own databases, like Elsevier, Thomson Reuters and Springer [4, 14, 15].

The search service of publishing company shows the clear author-organization relationship. If most authors have published journals or books via them, it is the best choice without any doubts. Unfortunately it is not the case for the participants of ASCO annual meeting.

In research area, CiteSeer and DBLP are widely used. They are free to public use and have more than 1 millions publications. Many author disambiguation algorithms are developed and tested against them, but they are not suitable for our system since they are focused on the computer science industry. CiteSeer is multi-disciplined but it is member-controlled.

Among all the medicine databases, Pubmed is the best. It comprises more than 21 million citations from conference, science journals and online books. Except a few latest abstracts which were just published or about to be published, we can almost find all the abstracts we are interested in.

We search each author of the abstract showed in Figure1-1 in all of the bibliographic databases mentioned above and list parts of the result in Table 3-8 (the complete result is list in Appendix A, Table A-1).

Database	Author	Results	Author	Results	Author	Results	Author	Results
CiteSeer	Caron WP	0	Lay JC	0	Fong AM	0	La-Beck NM	0
DBLP	Caron WP	0	Lay JC	0	Fong AM	2	La-Beck NM	0
MedlinePlus	Caron WP	0	Lay JC	5	Fong AM	0	La-Beck NM	0
pubMed	Caron WP	2	Lay JC	44	Fong AM	28	La-Beck NM	2

Table 3-8: Search Results for Author

3.3 Module Design

For the convenience of future extension and update, we try to design the whole system to be highly modular and structural. The system is designed to be of a three-layer structure, demonstrated in Figure 3-9. These three layers are: Support Layer, Communication Layer and Application Layer. Each layer has its own responsibilities and they all work together do finish the author-organization disambiguation task

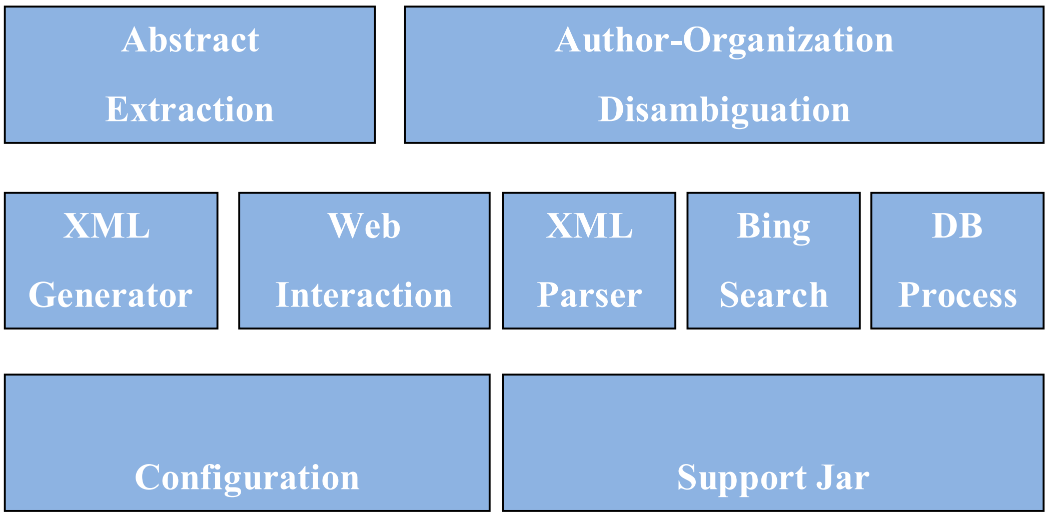


Figure 3-9: System Module Design

3.3.1 Support Layer

The support layer is at the bottom of the whole system. It is mainly responsible for dealing with the customised configuration and providing necessary support for upper layers. As other systems, our system also has a configuration file. This file contains system parameters, some of them are used to control the system's process and some are used to set the system environment.

3.3.2 Communication Layer

Communication layer is in the middle of the system. It accepts the parameters from Support Layer, communicates with the outer components and serves the Application Layer with different functions. It consists of several independent modules. Each of them corresponds to an outer component.

XML Generator and XML Parser module are to provide the format convert between the system internal data structure and standard XML format. With these two modules, our system becomes very flexible. On the one hand, any sample data can be handled by our system, as long as they are in XML format; on the other hand, the output of the result is in XML format so that it can be reviewed on any platform.

Web Interaction module provides a way for our system to access the web pages. When we want to access a web page, we just need to type the correct URL in the browser then wait it to bring back the contents. Our system is a desktop application, so no browser gets involved. We design the Web Interaction module which works almost like a browser. It accepts an URL and returns that page's content in HTML format as demonstrated in Figure 3-10.

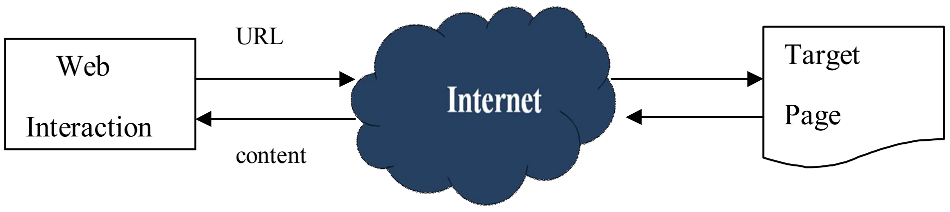


Figure 3-10: How Web Interaction Module works

DB Access module is an encapsulation class of database operation, including to create or to shut down the connection to a database, to execute a SQL query, to update the database etc. All the database operations are encapsulated in this module, so it is very easy for future maintenance and extension. For example, we are currently using MySQL, if we decide to use MS SQL one day, we just need to change this module and keep everything else untouched.

Since database operation is frequently used, we make the connection a class variable and encapsulate the open and close operation as well. A typical use is shown in Figure 3-11:

```
        Entry Modul

main( ){
    create a DB access instance, db;
    db.openConnection( );
    // other non-DB operations
   db.closeConnection( );
}
```

```
        Other Module

procedure( ){
     //wheneven want to do DB stuff
     //just use it since we have
    //create the connection in main
    db.certainOpertaion( );
}
```

Figure 3-11: Design of DB access

Bing Search module takes advantage of Bing API to do the search via Bing search engine. Given a search keyword, Bing API can retrieve the search results in 3 formats:

- XML
- JSON (JavaScript Object Notation) is a lightweight data-interchange texture format. Like XML, it is easy for human to read and write.
- SOAP (Simple Object Access Protocol) is a W3C standard protocol. It is designed for exchange of information in a decentralized, distributed environment.

After thinking of each format carefully, we finally decide to use JSON. In our system, we simply need to get information from Bing. There are no peer nodes, so no need to use SOAP. Comparing to XML, JSON is smaller because it does not have tags. Maybe a smaller size can not make the system faster if the system only uses the search once or twice, but in a system like ours which has to do the search frequently, this does matter.

3.3.3 *Application Layer*

Application Layer is on the top of the whole system. It is built on lower layers and implements the main logic process. It is comprised of two modules: extract module and author-organization disambiguation module.

These two modules essentially have nothing to do with each other – Figure 3-12 shows the input and output for each module. If you only want to extract abstracts from a specified web page, you can run Extraction module separately. If you have a well-formatted XML file and the XML format can be recognized by our system, you can apply Author-Organization disambiguation module to it. Of course you can use the output of Extraction module as the input of Author-Organization Disambiguation module, in which way you connect both modules together.

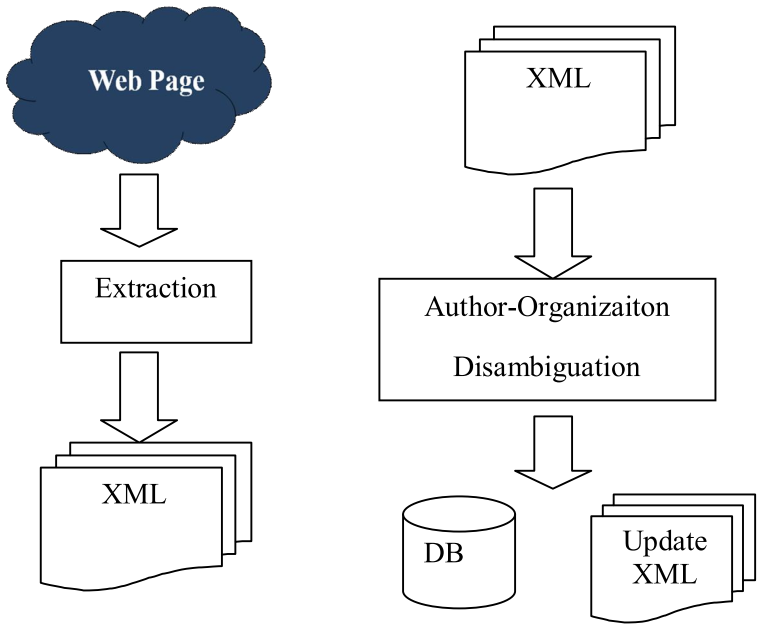
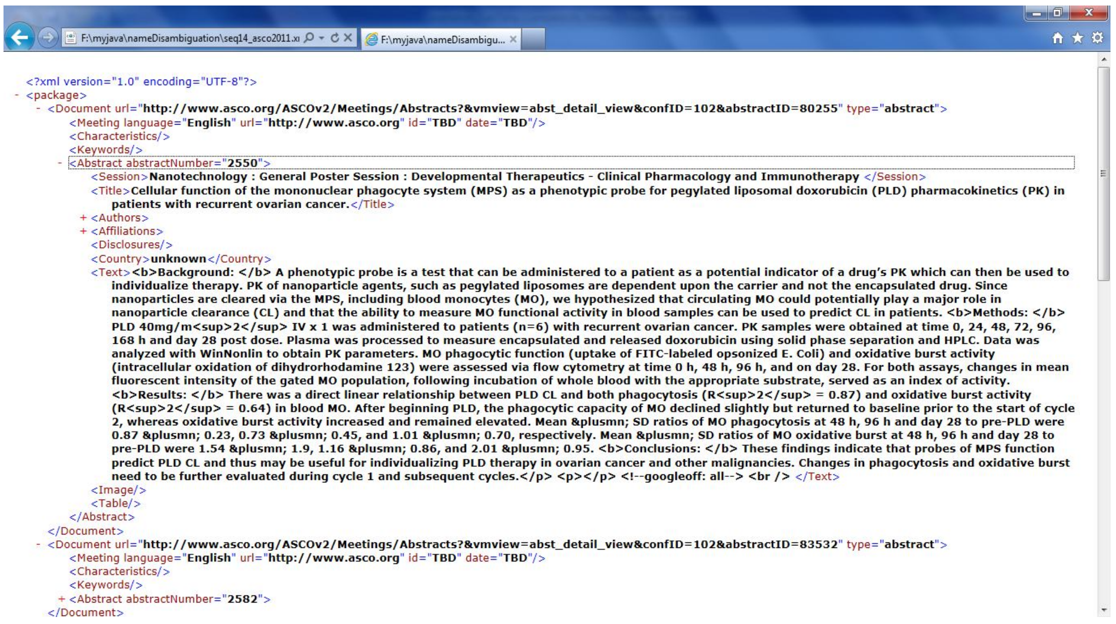


Figure 3-12: Design of Input and Output for Application Layer

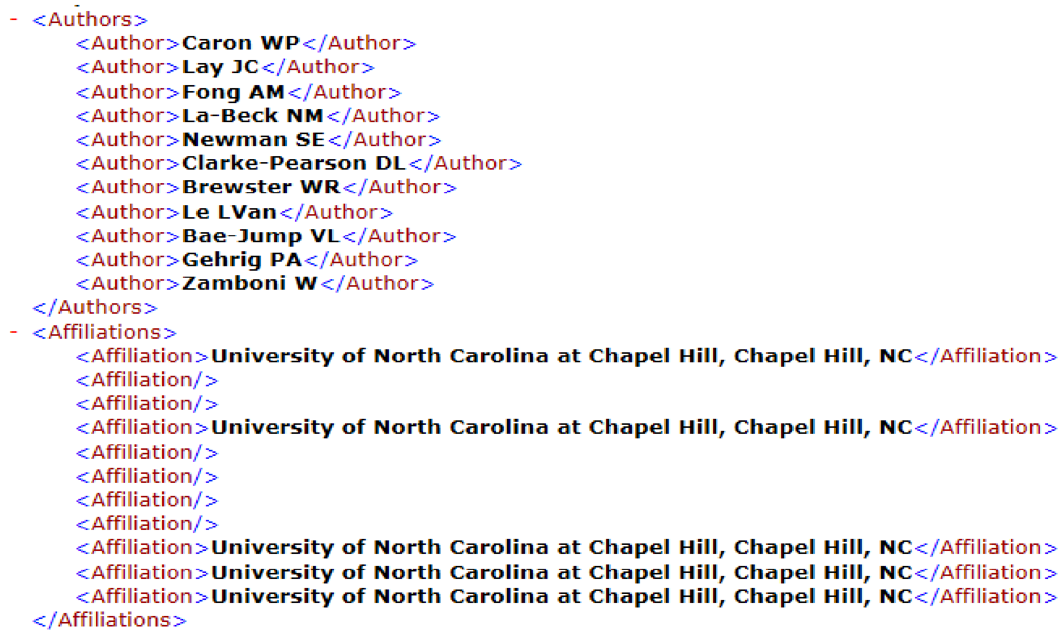
Abstract Extraction Module, as indicated by its name, is to extract abstracts and save it into a XML file. ASCO normally groups abstract according to the research field, so the extraction module will put the-same-field abstracts into a single XML file. Figure 3-13 shows how the XML should look. Each abstract is a <Document> node in xml file, which contains information like title, author, affiliation, abstract text, table even image.



```
<?xml version="1.0" encoding="UTF-8"?>
- <package>
  - <Document url="http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=80255" type="abstract">
      <Meeting language="English" url="http://www.asco.org" id="TBD" date="TBD"/>
      <Characteristics/>
      <Keywords/>
    - <Abstract abstractNumber="2550">
        <Session>Nanotechnology : General Poster Session : Developmental Therapeutics - Clinical Pharmacology and Immunotherapy </Session>
        <Title>Cellular function of the mononuclear phagocyte system (MPS) as a phenotypic probe for pegylated liposomal doxorubicin (PLD) pharmacokinetics (PK) in
           patients with recurrent ovarian cancer.</Title>
      + <Authors>
      + <Affiliations>
        <Disclosures/>
        <Country>unknown</Country>
        <Text><b>Background: </b> A phenotypic probe is a test that can be administered to a patient as a potential indicator of a drug's PK which can then be used to
           individualize therapy. PK of nanoparticle agents, such as pegylated liposomes are dependent upon the carrier and not the encapsulated drug. Since
           nanoparticles are cleared via the MPS, including blood monocytes (MO), we hypothesized that circulating MO could potentially play a major role in
           nanoparticle clearance (CL) and that the ability to measure MO functional activity in blood samples can be used to predict CL in patients. <b>Methods: </b>
           PLD 40mg/m<sup>2</sup> IV x 1 was administered to patients (n=6) with recurrent ovarian cancer. PK samples were obtained at time 0, 24, 48, 72, 96,
           168 h and day 28 post dose. Plasma was processed to measure encapsulated and released doxorubicin using solid phase separation and HPLC. Data was
           analyzed with WinNonlin to obtain PK parameters. MO phagocytic function (uptake of FITC-labeled opsonized E. Coli) and oxidative burst activity
           (intracellular oxidation of dihydrorhodamine 123) were assessed via flow cytometry at time 0 h, 48 h, 96 h, and on day 28. For both assays, changes in mean
           fluorescent intensity of the gated MO population, following incubation of whole blood with the appropriate substrate, served as an index of activity.
           <b>Results: </b> There was a direct linear relationship between PLD CL and both phagocytosis (R<sup>2</sup> = 0.87) and oxidative burst activity
           (R<sup>2</sup> = 0.64) in blood MO. After beginning PLD, the phagocytic capacity of MO declined slightly but returned to baseline prior to the start of cycle
           2, whereas oxidative burst activity increased and remained elevated. Mean &plusmn; SD ratios of MO phagocytosis at 48 h, 96 h and day 28 to pre-PLD were
           0.87 &plusmn; 0.23, 0.73 &plusmn; 0.45, and 1.01 &plusmn; 0.70, respectively. Mean &plusmn; SD ratios of MO oxidative burst at 48 h, 96 h and day 28 to
           pre-PLD were 1.54 &plusmn; 1.9, 1.16 &plusmn; 0.86, and 2.01 &plusmn; 0.95. <b>Conclusions: </b> These findings indicate that probes of MPS function
           predict PLD CL and thus may be useful for individualizing PLD therapy in ovarian cancer and other malignancies. Changes in phagocytosis and oxidative burst
           need to be further evaluated during cycle 1 and subsequent cycles.</p> <p></p> <!--googleoff: all--> <br /> </Text>
        <Image/>
        <Table/>
      </Abstract>
    </Document>
  - <Document url="http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=83532" type="abstract">
      <Meeting language="English" url="http://www.asco.org" id="TBD" date="TBD"/>
      <Characteristics/>
      <Keywords/>
    + <Abstract abstractNumber="2582">
    </Document>
```

Figure 3-13: Design of Extracted Abstract into XML Format

Author-Organization Disambiguation Module aims to reduce correct author-organization relationship by executing Two Stage disambiguation mentioned in Section 3.1. The results of this module will be updated into the XML file, as shown in Figure 3-14. Every author in <Authors> node should have a corresponding organization in <Affiliations> node. If the organization is found, it should be list; otherwise, <Affiliation/> will be disaplyed.



```
- <Authors>
    <Author>Caron WP</Author>
    <Author>Lay JC</Author>
    <Author>Fong AM</Author>
    <Author>La-Beck NM</Author>
    <Author>Newman SE</Author>
    <Author>Clarke-Pearson DL</Author>
    <Author>Brewster WR</Author>
    <Author>Le LVan</Author>
    <Author>Bae-Jump VL</Author>
    <Author>Gehrig PA</Author>
    <Author>Zamboni W</Author>
  </Authors>
- <Affiliations>
    <Affiliation>University of North Carolina at Chapel Hill, Chapel Hill, NC</Affiliation>
    <Affiliation/>
    <Affiliation/>
    <Affiliation>University of North Carolina at Chapel Hill, Chapel Hill, NC</Affiliation>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation>University of North Carolina at Chapel Hill, Chapel Hill, NC</Affiliation>
    <Affiliation>University of North Carolina at Chapel Hill, Chapel Hill, NC</Affiliation>
    <Affiliation>University of North Carolina at Chapel Hill, Chapel Hill, NC</Affiliation>
  </Affiliations>
```

Figure 3-14: Design of Updated XML File

4. System Implementation

Our system is not an isolated node. As discussed in Chapter 3.2, our system needs to interaction with other components. Each component has different services provided by different companies or organizations. We compared the advantage and disadvantage of each possible option and finally decided out system will work together with those components list in Figure 4-1.

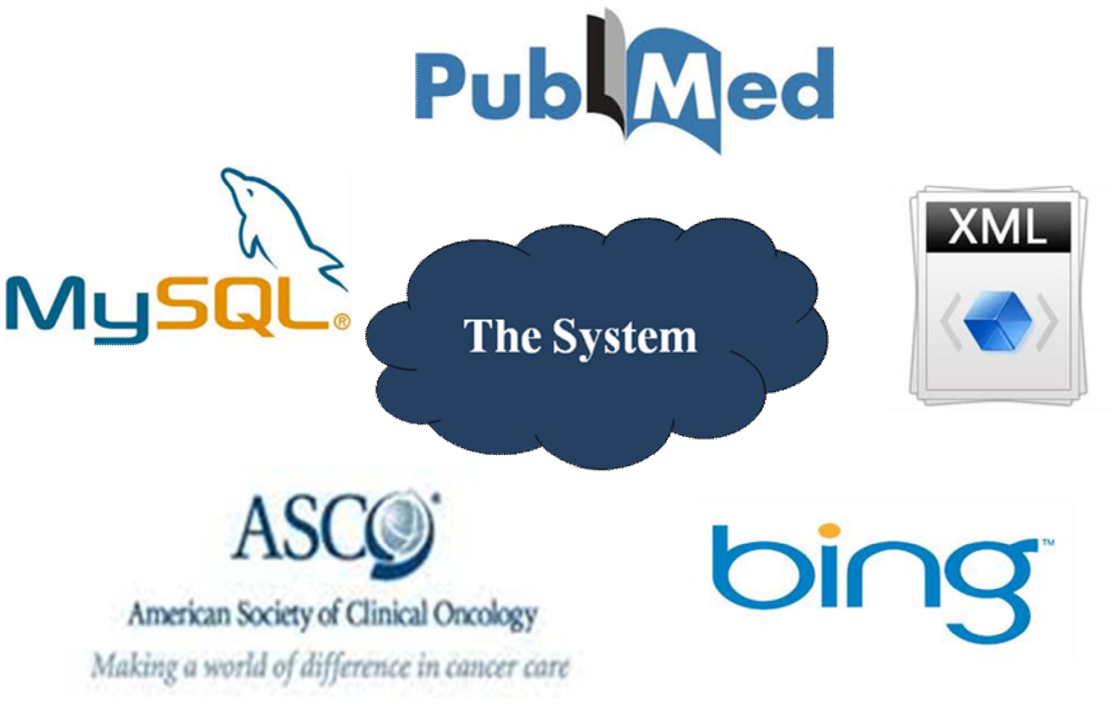

PubMed
MySQL
The System
XML
ASCO
American Society of Clinical Oncology
Making a world of difference in cancer care
bing

Figure 4-1: Component overview of the system

4.1 Co-author Based Author disambiguation

For the authors with the same names, we treat them as identical people if and only if they have direct or indirect co-author relationship. Given an author of one abstract, we firstly try to get abstracts with the identical names; next we check the co-author relationship and finish the author disambiguation. The basic algorithm can be described as below:

---

a) Assume we have an abstract P. Abstract P has a set of author A={A_1, A_2…A_n} (n >=1).

b) For each author A_i in A, we do the following steps:

    i) Create an empty List L, which is used to save all the abstracts written by author A_i.

Create an array C[ ], which is used to save the co-authors.

ii) Add all the items of A into C

iii) Search author A_i and get a set of abstracts P' = {P_1, P_2…P_k} (m>=0). If P' is empty, which means we fail to find any other abstracts for this author, jump to step b) to process the next author.

iv) For each abstract P_j in P', extract its author set A' = {A'_1, A'_2….A'_m}. A' must contain an author named A_i. To differentiate it from the A_i in step b), we mark this A_i as A'_i.

- If $C \cap A' - A_i \neq \Phi$ (it means author Ai has direct co-author with author A', thus we think they are actually referring to the same person), add P_j into L, add all the items of A' into C and jump to step iv) to process the next abstract.
- For each A_x in A' and each C_y in C, search if A_x has ever published with C_y. If any abstracts are found, it means A_i and A'_i have the indirect co-author relationship. So they are also referring to the same person. Add P_j into L, add all the items of A' into C and jump to step iii) to process the next abstract.

v) All the abstracts in L are believed written by the same author A_i.

How to judge the direct co-author relationship is easy to understand, so we only demonstrate the process of checking indirect co-author relationship. Assume author A and author A_1, A_2…A_n are co-authors for one abstract; Author A' and author B_1, B_2…B_m are co-authors for another different abstract. To check whether A' and A are actually the same person, according to indirect co-author algorithm, for each co-author of A we have to check whether he has written with any co-author of A'.

Figure 4-2 (a) –(c) describes the process. At first, we do a search for author A_1 and each co-author of A', as demonstrated in Figure 4-2 (a); if no co-author relationship is found, we do another search for A_2 and each co-author of A', as demonstrated in Figure 4-2 (b); such kind of search is executed repeatedly until the co-author relationship is found or the last author has been checked, like the Figure 4-2 (c).

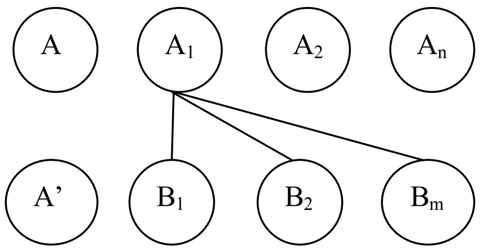


Figure 4-2 (a)

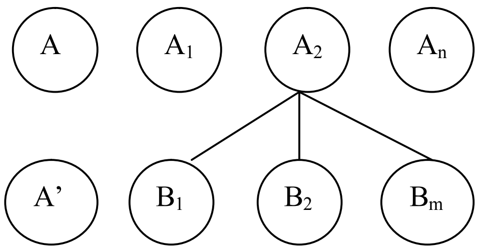


Figure 4-2 (b)

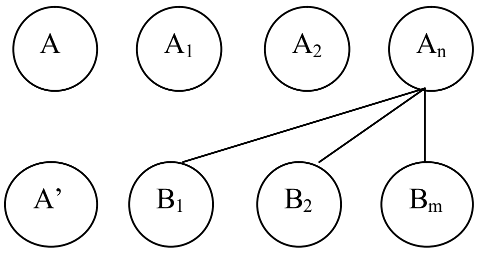


Figure 4-2 (c)

If any search during (a) – (c) returns an abstract, we think A' is the same author as A. So we put A' and all his co-authors into author A's co-author list. In this case, if we want to check another homonymy author A'', we have do more compares. Figure 4-3 shows that each co-author of A'' has to be compared with both A_i and B_i.

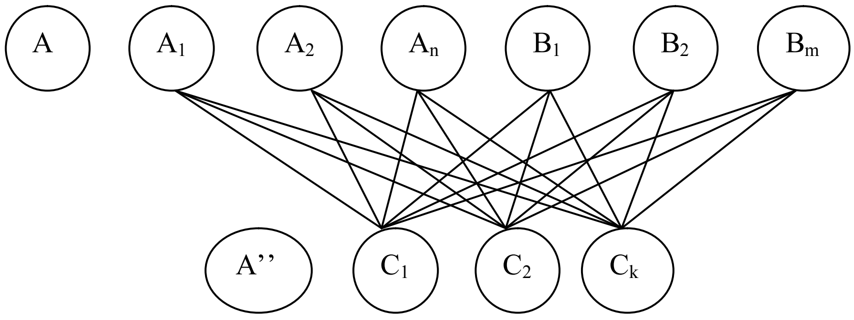


Figure 4-3: A's co-author list gets updated. Each will be checked with the co-authors of A''

This algorithm is simple and each to implement, but it will be harmful to the efficiency. Let us assume that A and A' are different people, so Figure 4-3 (a), (b) and (c) will be executed. That is, the system has to compare (n x m) times. Now we have another author A'' to be disambiguated. If one or several co-authors of A'' are the same as A', unnecessary duplicated compare will be done. Figure 4-4 shows how this inefficient thing happens.

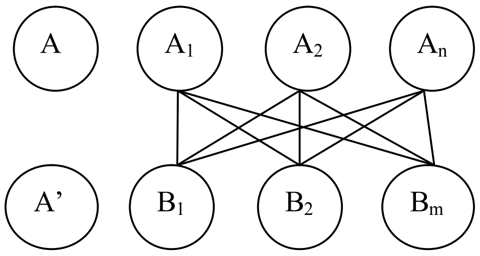


Figure 4-4 (a): After (n x m) compare, A and A' are proved different

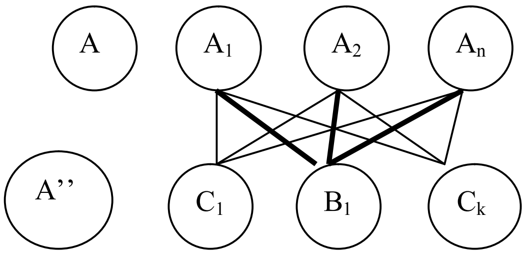


Figure 4-4 (b): Unnecessary duplicated compare to B_1, presented with bold line, because B_1 has been already checked in (a)

From Figure 4-4 (b), we know that the system will do n times of unnecessary compare for each duplicated co-author. In the research field, researchers are always in a small group so the number of unnecessary compare will become very large. It even can be over thousands depends on the scale of data set.

This problem happens because the system does not record the history compare, so it has no way to know whether the co-author it is going to compare has ever been checked or not. In other words, if the system remembers all the checked names and the results, it can jump to the conclusion about their relationship directly the next time when it meets the same name. We use database to preserve all the history data. So the author-disambiguation algorithm will be revised like this (revised parts are in Bold and Italic):

---

a) Assume we have an abstract P. Abstract P has a set of author $A=\{A_1, A_2 \ldots A_n\}$ $(n >= 1)$.

b) For each author A_i in A, we do the following steps:

   i) Create an empty List L, which is used to save all the abstracts written by author A_i.

   Create an array C[ ], which is used to save the co-authors.

   vi) Add all the items of A into C

   vii) Search author A_i and get a set of abstracts $P' = \{P_1, P_2 \ldots P_k\}$ $(m>=0)$. If P' is empty, which means we fail to find any other abstracts for this author, jump to step b) to process the next author.

   viii) For each abstract P_j in P', extract its author set $A' = \{A'_1, A'_2 \ldots . A'_m\}$. A' must contain an author named A_i. To differentiate it from the A_i in step b), we mark this A_i as A'_i.

      - If $C \cap A' - A_i \neq \Phi$ (it means author Ai has direct co-author with author A', thus we think they are actually referring to the same person), add P_j into L, add all the items of A' into C

        Save P and P_j into table COAUTHOR

        Jump to step iv) to process the next abstract.

      - For each A_x in A' and each C_y in C, do the following steps:

        - ***Search in table NON_COAUTHOR if A_x and C_y have been confirmed to be a non-coauthor relationship. If they are, process the next author pair.***

        - Search on web for A_x and C_y. If any abstracts are found, it means A_i and A'_i have the indirect co-author relationship. So add P_j into L, add all the items of A' into C

          save P and P_j into table COAUTHOR.

          Jump to step iv) to process the next abstract

- ***When we reach here, it means all the authors in P have no co-author relationship with P_j, so we record all the author pairs into NON_COAUTHOR table.***

ix) All the abstracts in L are believed written by the same author A_i.

4.2 Sole Information Compare

In Section 3.1.2, we mentioned one possible result is that clustered abstract contains clear author-organization relationship. This maybe happen when the data source only provides the organization for first author, i.e. Pubmed, or when the provider gives out the clear relationship, i.e. Spring Website. Sole Information Compare is designed to get the organization in this situation.

To sum up, the algorithm we used in this system is described as below:

```
a) Create an empty list L.
b) Get all the organizations from sample abstract, remove all the meaningless words,
   like "a", "the", "of", "for", "at" and other prepositions then put them into L.
c) For each clustered abstract, get its sole organization O. O = <part 1> <separator>
   <part 2> <separator>…<part n><separator>
    i)   remove all the meaningless words from O
    ii)  Split O into groups according to the separator and save the result into a array
         G[ ]. G[ ] = { part 1, part 2… part n}
    iii) For each organization O' in L
           • Split O' into groups G'[ ].
           • Compare G[ ] and G'[ ] to check if one of them is equal to or
             contained by the other. If it is, current O' is what we are looking for.
             So return O' and end the program.
    iv)  No organization is found matched, return NULL and end the program.
```

4.3Intersection Compare

Intersection Compare applies when the clustered abstracts have uncertain organization information for that author. That is, the clustered abstracts contain a set of authors and a set of their organizations without one-to-one mapping, just like the sample abstract.

Before we describe the algorithm, let us consider the simplest example. Assume we have a sample abstract which contains author set $A=\{A_{author}, A_1, A_2\}$ and organization set $O=\{O_1, O_2, O_{author}\}$. We also have a clustered abstract which contains author set $A'=\{A'_{author}, A'_1, A'_2\}$ and organization set $O=\{O'_1, O'_{author}, O'_3\}$. If $A \cap A' = \{A_{author}\}$, $O \cap O'$ should get $\{ O_{author}\}$ which is the organization we are looking for.

The simplest example mentioned above assumes that each author has different organization. In this real world, it is quite common that some authors of an abstract work in the same organization or one author writes with different people from the same organization. This means the intersection result of two organization sets may contains more than one element. In theory, as long as we repeat the intersection, we will at last get an empty set which means no organization is found or the right organization.

The algorithm can be described as below:

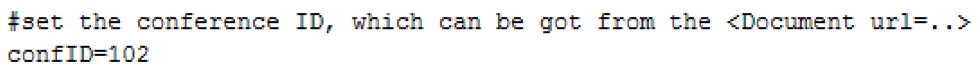
```
a) Given an abstract, create an author list A and organization list O.
b) Create a copy list for A and O:  $A_{tmp} = A$ and $O_{tmp} = O$.
c) For each abstract P in the clustered abstract, do the following
    i) Extract the author list A' and organization list O'
    ii) Do the intersection operation and save the result:
        $A_{tmp} = A_{tmp} \cap A'$;  $O_{tmp} = O_{tmp} \cap O'$
    iii) Check the result
        • If $A_{tmp}$ contains only one author and $O_{tmp}$ contains only one organization,
          that organization is what we are looking for. So return the organization
          and end the program.
        • If $O_{tmp} = \Phi$, it means no organization is found. End the program.
        • Otherwise, go back to step c) and process the next abstract.
```

4.4 Configuration

To make our system more flexible, we use a plain text file named ‘config.txt’ to keep system parameters. This file will be parsed and executed during the initialization of the program. In order to make the change effective, the system must be restarted. In section, we will firstly introduce the format of configuration file, by doing this one can easily edit or create the new configuration file; next we will introduce two important system parameters.

4.4.1 Configuration File

Configuration file is in a Tag and Value format. The advantages are: a) it is human readable and can be well understood even by those who does not have special knowledge on computer; b) system can parse this file with very simple code. A configuration File may have the data like in Figure 4-5:

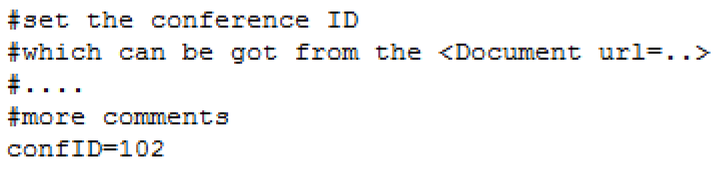


```
#set the conference ID, which can be got from the <Document url=..>
confID=102
```

Figure 4-5: Fragment of config.txt

Figure 4-5 shows the typical format of a configuration file. In the configuration file, we treat each line as a meaningful unit. Only three kinds of lines are allowed:

- An empty line which means nothing. Usually it is used as separator.
- A comment line. A comment line must start with character “#”. All the contents after “#” will be treated as comments. You can put the comment in one line or in multiple lines, like Figure 4-6.

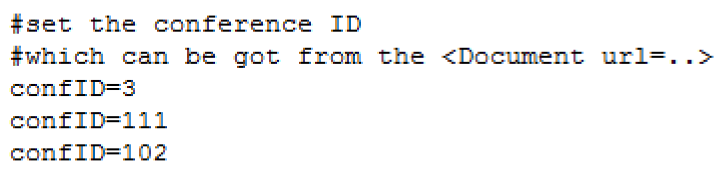


```
#set the conference ID
#which can be got from the <Document url=..>
#....
#more comments
confID=102
```

Figure 4-6: Comments in Multiple Lines

- A setting line. This line has a strict format as “<tag> = <value>”. i.e. the tag in Figure4-6 is ‘confID’ and its value is ‘102’. When the parsing module read a setting line, it treats all the letters before “=” as the name of tag and all the letters after “=” as the value. So space before and after “=” does not matter. Thus “confID=102” is the same as “confID   = 102”.

  For the same configuration tag, more than one setting line is allowed. But the latter one will override the previous value, so at the end the value is actually the value of the last occurrence. As shown in Figure 4-7, the value of ‘confID’ is eventually ‘102’.



```
#set the conference ID
#which can be got from the <Document url=..>
confID=3
confID=111
confID=102
```

Figure 4-7: Multiple setting line for one tag

4.4.2 System Parameter: shortenInitial

As discussed in Chapter 3, the author-disambiguation algorithm requires the system to search the name in web. Our system does not bother to fix the spelling problems, like spelling mistakes and the different spelling format due to the cultural reasons. But we do need to consider the middle name. Middle names may be included in the name initial and may not. Author name "Whitney P Caron" can be published with two different format: "Caron WP" and "Caron W". For the reference convenience, we call the previous format 'full initial name' and the latter 'shorten initial name'.

In most cases a full initial name is provided, so our system will search the full initial name anyway. Besides, a system parameter shortenInitial is provided. This parameter is to control whether a name initial without middle name is used or not by the system.

This parameter can be configured in configuration file. It will allow the system to search shorten initial name, if and only if it is set to be 'true'; all the other values including 'yes' will let it treated as false. Table 4-1 shows what name is to be used as the search name depends on the value of shortenInitial

Value of shortenInitial	**Author Name**	**Search String**
true	Whitney.P Caron	"Caron WP"[First Author]
Other values	Whitney.P Caron	"Caron WP"[First Author] or "Caron W"[First Author]

Table 4-1: shortenInitial and the Search String

An intuitive judgement about how this parameter will affects the system is that it will bring more returned results. It is out of question that it will take the system more time to do the author-disambiguation when the parameter is true. But will it help to improve the performance as well? We will live the question to Chapter 5 and 6. Now we make the following statement:

Note:

In the following sections, shortenInitial is set to be false by default unless especial statement.

4.4.3 System Parameter: maxMatchedCnt

In Chapter 3, we talked about the co-author based author-disambiguation algorithm. To test the co-author relationship, we have to search the author pair in the web. Based on the algorithm, once we have found a matching abstract, we will add all the co-authors into the origin co-author list. This can ensure the correctness, but bring a problem. The co-author list becomes larger and larger, and the times of compare become larger and larger as well– called as compare problem.

Assuming each abstract has 5 co-authors and each is written by the same author:

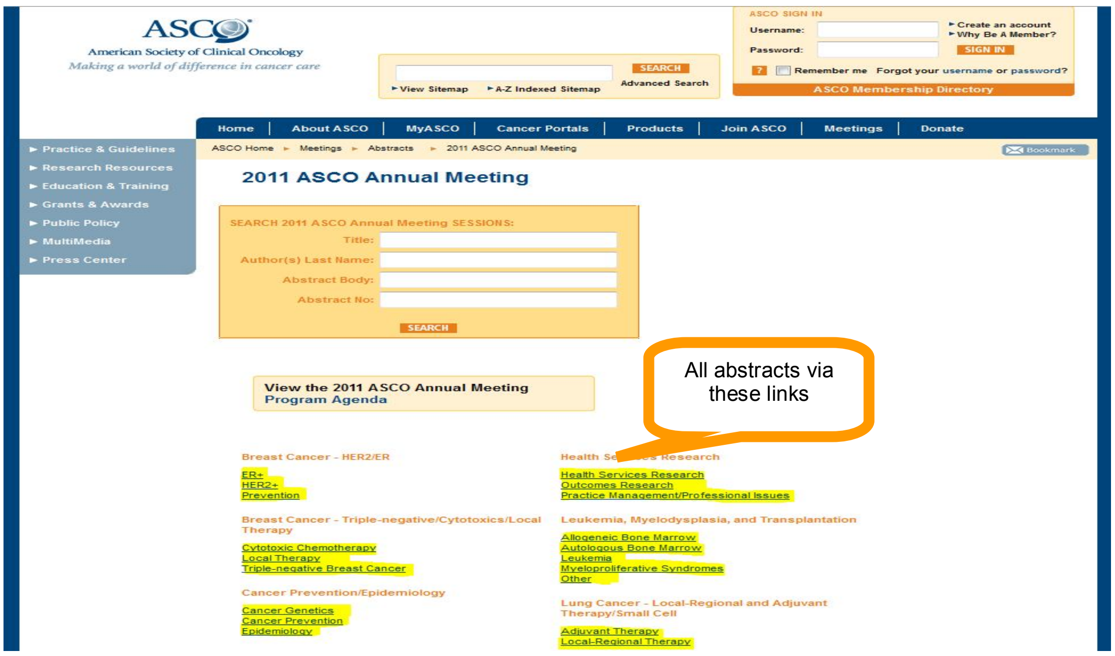
At the beginning:	*5 Co-authors in the list (from the sample abstract)*
Check abstract 1:	*Compare 5 x 5 = 25 times.* *Add abstract 1's co-author into list, so the list has 10 co-authors*
Check abstract 2:	*Compare 5 x 10 = 50 times.* *Add abstract 2's co-author into list, so the list has 15 co-authors*
	…
Check abstract n:	*Compare 5 x (5 x n) = 25n times.* *Add abstract n's co-author into list, so the list has 5 x (n + 1) co-authors*

To solve the problem, we introduce the parameter maxMatchCnt, which can be configured in configuration file. Parameter maxMatchCnt is used to control at most how many abstracts, which have already been confirmed to be written by the same author, are going to be used to confirm the organization. Taking above abstracts for example, if maxMatchCnt is set to be 1, the process will be end after finishing check abstract 1; if maxMatchCnt is set to be 2, it will be end after finishing check abstract 2.

4.5 Abstract Extraction

Abstract Extraction is one of our core modules. Its function is very clear: to extract all the abstract from ASCO website and convert it into XML file. The basic idea is firstly to send a request to ASCO website, then bring back the abstract and lastly convert it into XML.

Every year, ASCO will publish all the abstracts on its website. Those abstracts are put into different categories as shown in Figure 4-8. Each Category has several sub-categories, i.e. the first category on the top-left "Breast Cancer – HER2/ER" consist of "ER+", "HER2+" and "Prevention" sub-category.



2011 ASCO Annual Meeting

SEARCH 2011 ASCO Annual Meeting SESSIONS:
Title:
Author(s) Last Name:
Abstract Body:
Abstract No:
SEARCH

View the 2011 ASCO Annual Meeting
Program Agenda

Breast Cancer - HER2/ER
ER+
HER2+
Prevention

Breast Cancer - Triple-negative/Cytotoxics/Local Therapy
Cytotoxic Chemotherapy
Local Therapy
Triple-negative Breast Cancer

Cancer Prevention/Epidemiology
Cancer Genetics
Cancer Prevention
Epidemiology

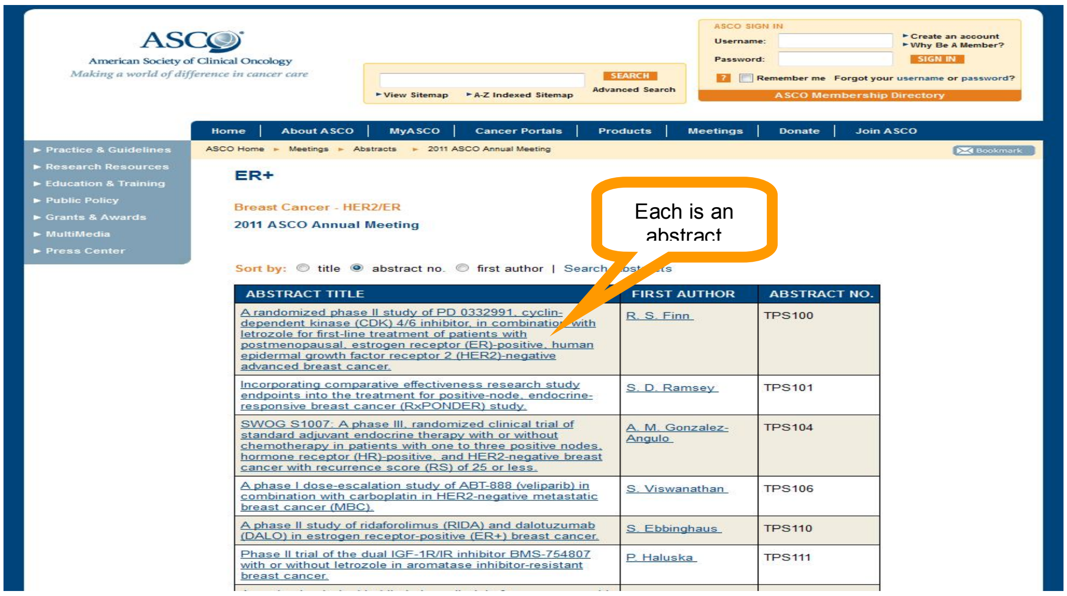
Health Services Research
Health Services Research
Outcomes Research
Practice Management/Professional Issues

Leukemia, Myelodysplasia, and Transplantation
Allogeneic Bone Marrow
Autologous Bone Marrow
Leukemia
Myeloproliferative Syndromes
Other

Lung Cancer - Local-Regional and Adjuvant Therapy/Small Cell
Adjuvant Therapy
Local-Regional Therapy

Figure 4-8: Abstract Category

Every sub-category leads to another web page, which list all the abstracts' titles as shown in Figure 4-9. The number of abstracts contained by one sub-category is quite different from each other: from less than 10 abstracts to more than 100.



ER+

Breast Cancer - HER2/ER
2011 ASCO Annual Meeting

Sort by: ○ title ● abstract no. ○ first author | Search abstracts

ABSTRACT TITLE	FIRST AUTHOR	ABSTRACT NO.
A randomized phase II study of PD 0332991, cyclin-dependent kinase (CDK) 4/6 inhibitor, in combination with letrozole for first-line treatment of patients with postmenopausal, estrogen receptor (ER)-positive, human epidermal growth factor receptor 2 (HER2)-negative advanced breast cancer.	R. S. Finn	TPS100
Incorporating comparative effectiveness research study endpoints into the treatment for positive-node, endocrine-responsive breast cancer (RxPONDER) study.	S. D. Ramsey	TPS101
SWOG S1007: A phase III, randomized clinical trial of standard adjuvant endocrine therapy with or without chemotherapy in patients with one to three positive nodes, hormone receptor (HR)-positive, and HER2-negative breast cancer with recurrence score (RS) of 25 or less.	A. M. Gonzalez-Angulo	TPS104
A phase I dose-escalation study of ABT-888 (veliparib) in combination with carboplatin in HER2-negative metastatic breast cancer (MBC).	S. Viswanathan	TPS106
A phase II study of ridaforolimus (RIDA) and dalotuzumab (DALO) in estrogen receptor-positive (ER+) breast cancer.	S. Ebbinghaus	TPS110
Phase II trial of the dual IGF-1R/IR inhibitor BMS-754807 with or without letrozole in aromatase inhibitor-resistant breast cancer.	P. Haluska	TPS111

Figure 4-9: List of Abstract Title under a Sub-category

The only way to get the details of one abstract is: firstly, go to the entry page (in Figure 4-8); secondly, get to the abstract list (in Figure 4-9) by clicking one of the sub-categories; lastly, click the abstract title to get the details. "Automated" requires our system to retrieve all the abstracts without human's interference. The ideal process should be like this: given an URL of the entry page, the system can recognize all the categories; then it accesses each category to get the list of abstracts; next it visits each detailed abstract, extracts information required and saves them into XML file. Given an URL of entry page, our system can extract all the abstracts automatically.

Our customer expect to extract the following information (please refer to Figure 4-10) from an abstract, so this system should be able to retrieve them all.

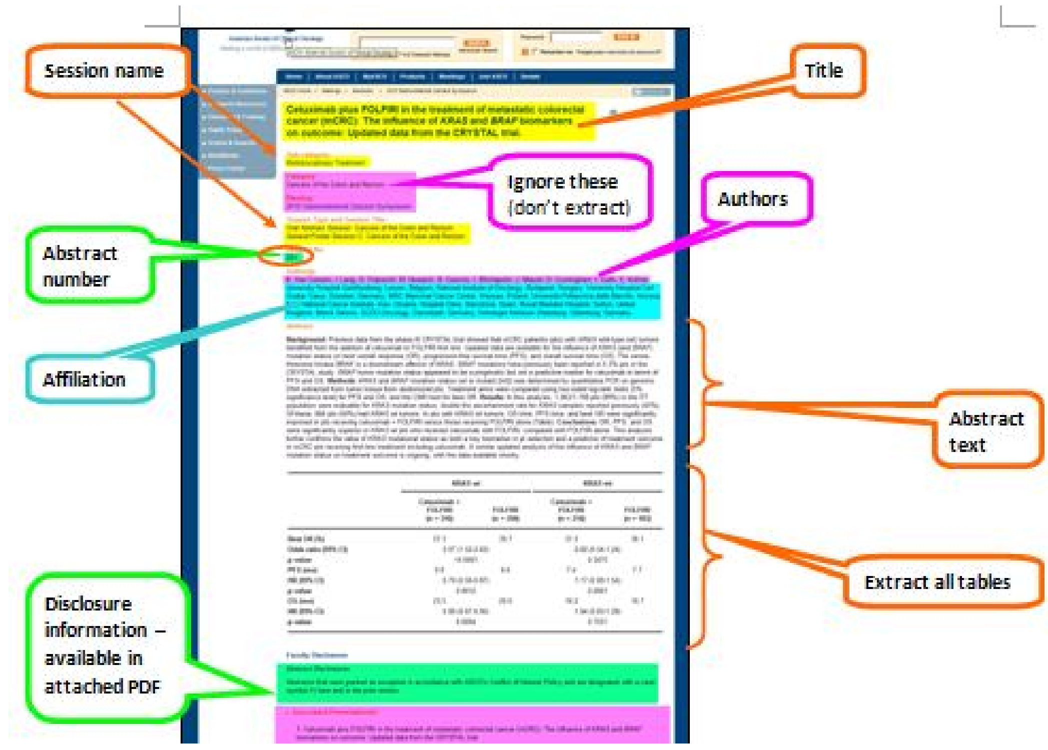


Figure 4-10: Abstract Extraction Requirements

Another thing we need to pay attention to is that the format of abstract may be different from year to year. For example, section "Session Type and Session Title" uses "," as separator in 2011, but it uses ";" instead in 2010.

By checking the page element with the firebug, we know exactly how to extract these items

- Title: marked by HTML element <h2> and </h2> under <table class="nodecontent">.
- Session Name: this part consists of two parts. One part is marked by <a> and </a> in <div> Sub-category </div>; the other is plain texture content after <div> Session Type and Session Title </div>
- Abstract Number: the plain texture content after <div> Abstract No </div>
- Author: it is plain texture content after <div> Author(s): </div>, from the beginning to the first occurrence of semicolon ";". Each author is followed by a comma ",".
- Affiliation: the plain texture content after <div> Author(s): </div>, from the first occurrence of semicolon ";" to the end. Each affiliation is followed by a semicolon
- Abstract Text: all plain texture between <p class="orangebold2"> Abstract </p> and <p class="orangebold">, except Table information and Image information
- Table: embedded in Abstract Text part and marked by <table> and </table>, if it exists

- Image: embedded in Abstract Text part and marked by <image> and </image>, if it exists
- Disclosure: marked by <div>Disclosure</div>, if it exists

4.6 Bing Search

Besides the medical database Pubmed, we try to take advantages of Bing search engine. Due to the lack of enough information, the system may fail to reduce the correct author-organization relationship. As one of the biggest and the most famous search engine, Bing probably can bring the information we need.

When doing a search via search engine, the search string is very important. A good search string can bring back the most relevant results, while a bad one may bring back loads of useless rubbish.

We tried different search string and list how many results are returned in Figure 4-11. Using the author name is not wise because too many results are there and most of them are from social network like Facebook, LinkedIn etc. Using name and a single keyword reduces the amount sharply, but still contains too many information.

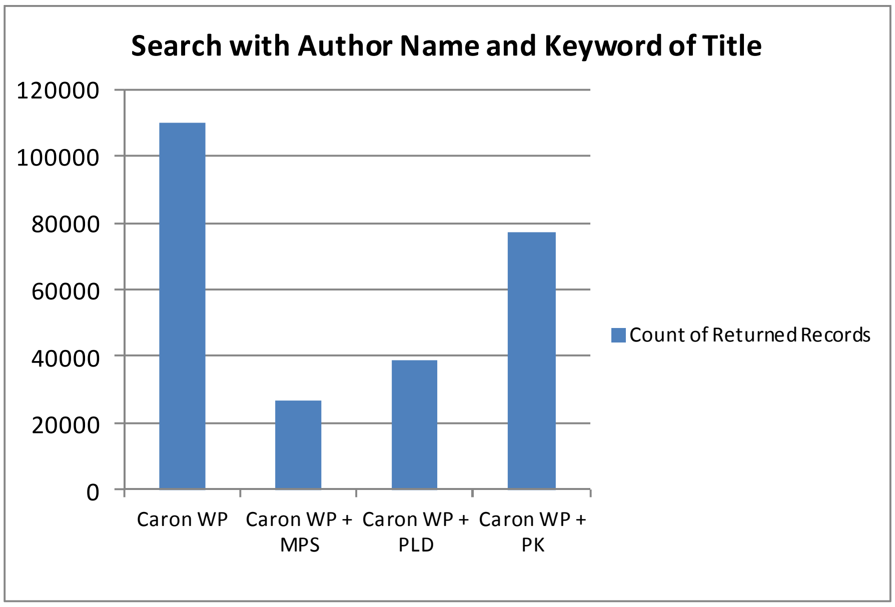


Figure 4-11: Returned results with different search keyword

We tried the name with more than one keyword because we thought it will narrow down the search range, thus retrieved less information. Unfortunately it is not always true. Figure 4-11 shows the returned results are even more than those with single keyword.

After a brief investigation, it turns out:

a) Bing uses OR not AND if several keywords are provides. The most front returned information is the most relevant, but the OR operation may ruin it. A living proof is that author name with all three keywords returned the most records, the last column in Figure 4-12

b) Bing may use other keywords to replace the origin one if it fails to find the matched keyword. One of the origin keyword is 'MPS', but most returned results are actually about MP3.

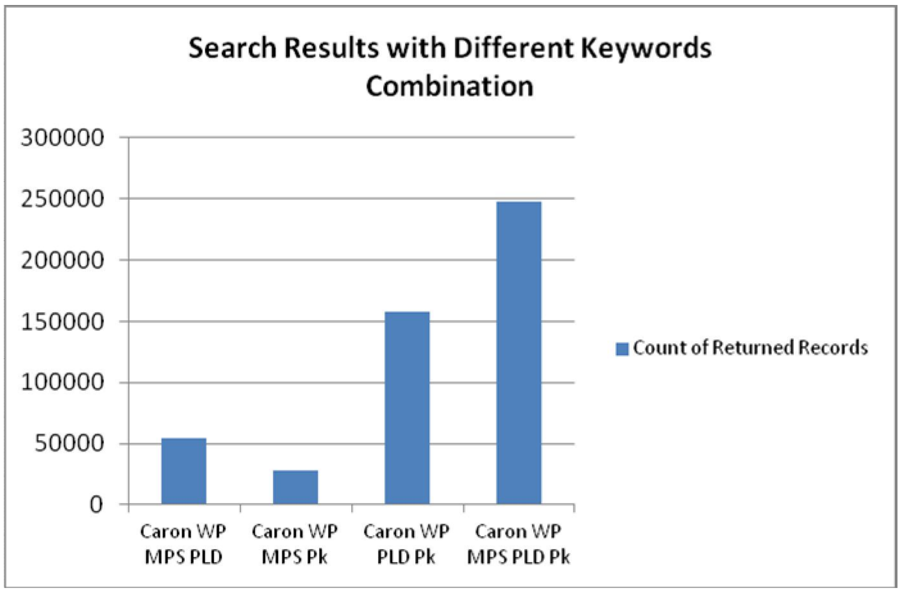


Figure 4-12: Returned results with different combinationkeyword

Even the returned results are highly relevant, it is also difficult for our system to handle it. Each result is a link to other web page, we can get back the information we want if we do not know the structure of that page.

One possible way is to build a database about web page's structure. We do not need to cover every single web page. Instead, we can focus on those which contain mainly the abstract information or other publishing information, like the big publish company.

Summary

Using Bing search has to solve two problems. One is how to decide the appropriate search string. Since Bing retrieves information around the whole world, no common rules can be applied for every situation. The other is how to get the information from variant web pages with their page structure unknown.

As for building the database of structure, it is in theory feasible but beyond our system's range, for our system currently is on prototype stage. However, for the convenience of future extension, we develop the Bing search module and leave it into this system.

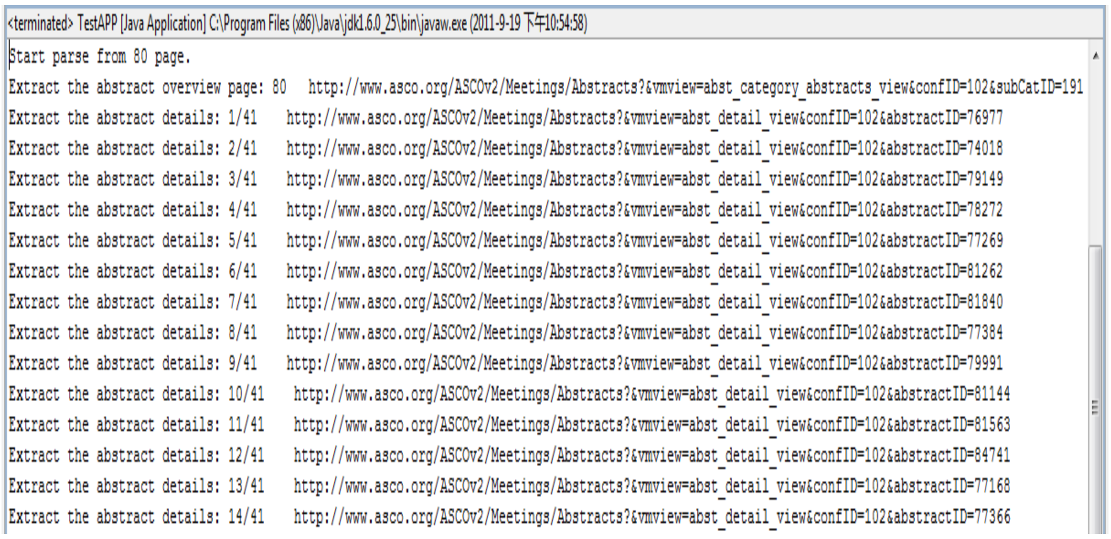
5. Test Results

Testing is very import because it can help us to check whether the system performs as we expect. Moreover it can help us to estimate the system. In this chapter, we will give some test results for each

5.1 Abstract Extraction

Test proves that our system can finish the whole extraction automatically. Figure 5-1 shows the run-time console, from which we can know following things:

- Which sub-category is being handled. "…abstract overview page: 80" means the system is processing the 80th sub-category.
- Which abstract is being extracted. "…abstract details: 1/41" means under the 80th sub-category, there are 41 abstracts and the system is processing the 1st one.



```
<terminated> TestAPP [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_25\bin\javaw.exe (2011-9-19 下午10:54:58)
Start parse from 80 page.
Extract the abstract overview page: 80   http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_category_abstracts_view&confID=102&subCatID=191
Extract the abstract details: 1/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=76977
Extract the abstract details: 2/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=74018
Extract the abstract details: 3/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=79149
Extract the abstract details: 4/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=78272
Extract the abstract details: 5/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=77269
Extract the abstract details: 6/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=81262
Extract the abstract details: 7/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=81840
Extract the abstract details: 8/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=77384
Extract the abstract details: 9/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=79991
Extract the abstract details: 10/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=81144
Extract the abstract details: 11/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=81563
Extract the abstract details: 12/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=84741
Extract the abstract details: 13/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=77168
Extract the abstract details: 14/41    http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=77366
```

Figure 5-1: Run-time Console Window

We run the extraction module for ASCO Meeting 2011, which contains 80 sub-categories in total. If given the URL of the entry page, the system can go through all the 80 sub-categories and its abstract lists.

5.2Co-author Based Author Disambiguation

We do the search with the name "Caron WP" and its short initial name "Caron W" and get 9 abstracts in total. By manual search and analysis, three of them are exactly the "Caron WP" of the sample abstract, which has the 'Y' flag in column "Same Person" in Table 5-1. Column "Result" is how our system thinks they are: "Y" means our system thinks they are the same person, while "N" means they are not.

No.	Title	Author Name	Same Person	Result
1	Allometric scaling of pegylated liposomal anticancer drugs.	Caron WP	Y	Y
2	Prevention strategies for antimicrobial resistance: a systematic review of the literature.	Caron WP	Y	N
3	An evidence-based elective on dietary supplements.	Caron W	Y	N
4	Family boundary ambiguity predicts Alzheimer's outcomes.	Caron W	N	N
5	Reducing caregiver burden: a randomized psychoeducational intervention for caregivers of persons with dementia.	Caron W	N	N
6	A biopsychosocial model for youth obesity: consideration of an ecosystemic collaboration.	Caron W	N	N
7	A biopsychosocial perspective on behavior problems in Alzheimer's disease.	Caron W	N	N
8	The Family Stories Workshop: stories for those who cannot remember.	Caron W	N	N
9	Predictors of depression in caregivers of dementia patients: boundary ambiguity and mastery.	Caron W	N	N

Table 5-1: Author Disambiguation Results

Sample abstract in Figure 1-1 contains the following information:

Title:	*Cellular function of the mononuclear phagocyte system (MPS) as a phenotypic probe for pegylated liposomal doxorubicin (PLD) pharmacokinetics (PK) in patients with recurrent ovarian cancer.*
Authors:	*W. P. Caron, J. C. Lay, A. M. Fong, N. M. La-Beck, S. E. Newman, D. L. Clarke-Pearson, W. R. Brewster, L. Van Le, V. L. Bae-Jump, P. A. Gehrig, W. Zamboni*
Organization:	*University of North Carolina at Chapel Hill, Chapel Hill, NC; Texas Tech University, Abilene, TX*

We run the author disambiguation for it and try to cluster each author. Table 5-2 shows the result. Column “Number of Identical Name” shows how many authors with the same name are found. Column “Number of Y/Y” shows home many abstracts are believed by our system to be written by the author and it is true. “Number of N/N” shows home many abstracts are believed by our system not to be written by the author and it is true. All the other cases fall into column “Number of Y/N”

Author	Number of Identical Name	Number of Y/Y	Number of N/N	Number of Y/N
W.P. Caron	2	1	0	1
J.C. Lay	0	0	0	0
A.M. Fong	6	4	2	0
N.M La-Beck	1	1	0	0
S. E. Newman	16	0	13	3
D. L. Clarke-Pearson	16	10	4	2
W. R. Brewster	6	4	1	1
L. Van Le	0	0	0	0
V. L. Bae-Jump	5	5	0	0
P. A. Gehrig	11	10	0	1
W. Zamboni	18	15	1	2

Table 5-2: Author disambiguation results

5.3 Organization Confirmation

We randomly pick up an abstract and run the module to test whether our system works well. Figure 5-2 list one organization of ASCO and Pubmed for the same author 'Caron WP'. Our system thinks they are the same organization

ASCO:	*University of North Carolina at Chapel Hill, Chapel Hill, NC*
Pubmed:	*Division of Pharmacotherapy and Experimental Therapeutics, Eshelman School of Pharmacy, University of North Carolina at Chapel Hill, 120 Mason Farm Road, Suite 1013, CB 7361, Chapel Hill, NC*

Figure 5-2: Example of different format

We test this algorithm with name 'M.Di Seri'. We choose this author because it has no record in Pubmed database so far, but has 4 abstracts in ASCO. The main information about these 4 abstracts are:

Abstract 1:
http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=78285
M.Di Seir and other 19 co-authors.

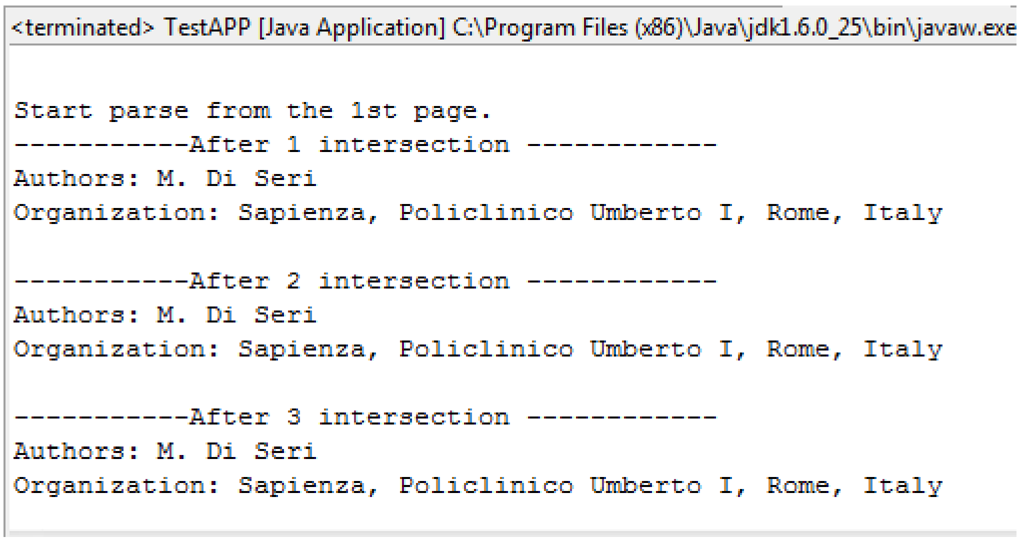
Abstract 2:
http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102

&abstractID=83496
M.Di Seir and other 10 co-authors.

Abstract 3:
http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=82701
M.Di Seir and other 8 co-authors.

Abstract 4:
http://www.asco.org/ASCOv2/Meetings/Abstracts?&vmview=abst_detail_view&confID=102&abstractID=87173
M.Di Seir and other 9 co-authors.

Figure 5-3 is the runtime window of executing the algorithm. The conclusion from our system is that author 'M.Di Seri' works in Sapienza, Policlinico Umberto I, Rome, Italy, which is correct.



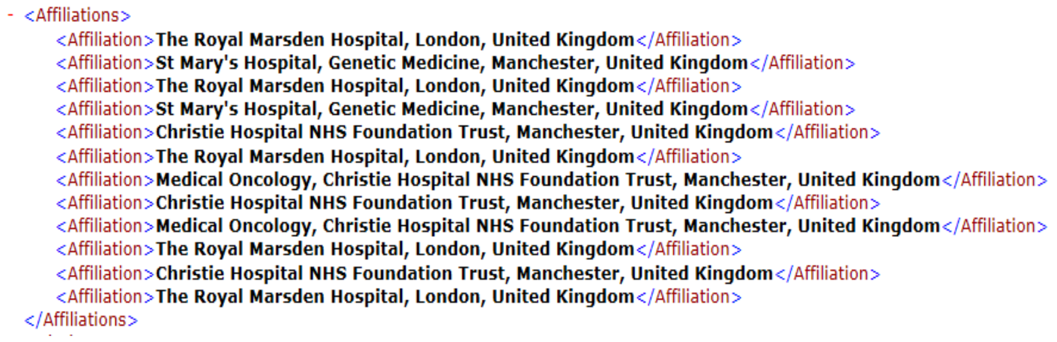
```
<terminated> TestAPP [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_25\bin\javaw.exe


Start parse from the 1st page.
-----------After 1 intersection ------------
Authors: M. Di Seri
Organization: Sapienza, Policlinico Umberto I, Rome, Italy

-----------After 2 intersection ------------
Authors: M. Di Seri
Organization: Sapienza, Policlinico Umberto I, Rome, Italy

-----------After 3 intersection ------------
Authors: M. Di Seri
Organization: Sapienza, Policlinico Umberto I, Rome, Italy
```

Figure 5-3: Runtime window of intersection compare

We force the program to do the intersection with all these 4 abstracts to observe the middle results after each intersection operation. In the real system, we should end it once only one organization left. The whole process is about texture compare, so the execution is very fast. The total execution time including sending web request to get back each abstract is only 3 minutes.

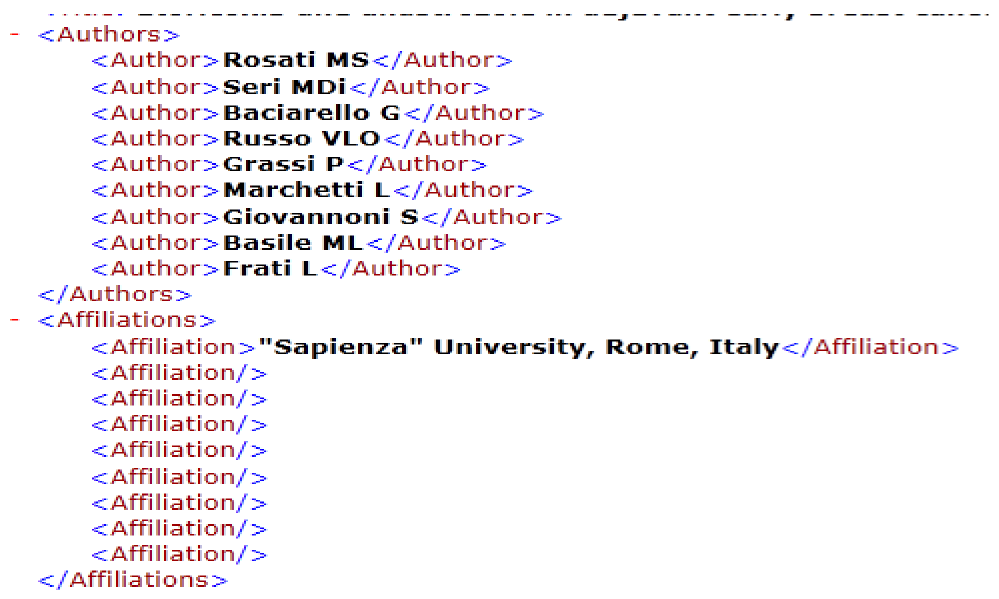
5.4 Relationship Disambiguation

So far, all the tests are put on small scale data. It is not enough to reach a conclusion. To give an impartial evaluation on the system's performance, we run the program with maxMatchedCnt=2 on hundreds of authors. We say the result is impartial, because the results contain very good result like in Figure 5-4 and very bad result like in Figure 5-5.



```
- <Affiliations>
    <Affiliation>The Royal Marsden Hospital, London, United Kingdom</Affiliation>
    <Affiliation>St Mary's Hospital, Genetic Medicine, Manchester, United Kingdom</Affiliation>
    <Affiliation>The Royal Marsden Hospital, London, United Kingdom</Affiliation>
    <Affiliation>St Mary's Hospital, Genetic Medicine, Manchester, United Kingdom</Affiliation>
    <Affiliation>Christie Hospital NHS Foundation Trust, Manchester, United Kingdom</Affiliation>
    <Affiliation>The Royal Marsden Hospital, London, United Kingdom</Affiliation>
    <Affiliation>Medical Oncology, Christie Hospital NHS Foundation Trust, Manchester, United Kingdom</Affiliation>
    <Affiliation>Christie Hospital NHS Foundation Trust, Manchester, United Kingdom</Affiliation>
    <Affiliation>Medical Oncology, Christie Hospital NHS Foundation Trust, Manchester, United Kingdom</Affiliation>
    <Affiliation>The Royal Marsden Hospital, London, United Kingdom</Affiliation>
    <Affiliation>Christie Hospital NHS Foundation Trust, Manchester, United Kingdom</Affiliation>
    <Affiliation>The Royal Marsden Hospital, London, United Kingdom</Affiliation>
  </Affiliations>
```

Figure 5-4: Example of good result



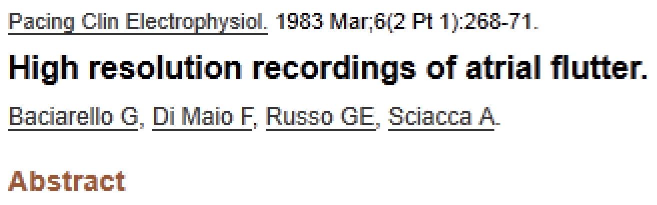
```
- <Authors>
    <Author>Rosati MS</Author>
    <Author>Seri MDi</Author>
    <Author>Baciarello G</Author>
    <Author>Russo VLO</Author>
    <Author>Grassi P</Author>
    <Author>Marchetti L</Author>
    <Author>Giovannoni S</Author>
    <Author>Basile ML</Author>
    <Author>Frati L</Author>
  </Authors>
- <Affiliations>
    <Affiliation>"Sapienza" University, Rome, Italy</Affiliation>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
    <Affiliation/>
  </Affiliations>
```

Figure 5-5: Example of bad result

This failure to get a clear author-organization relationship has several reasons: it might be because of the limitation of the algorithm or because of the lack of fact data. After testing on over thousands authors, we summarized the following reasons:

- No record is found, like author 'M.Di Seri' . The fact is that this author is a new research. Manual search shows that his first publication was in 2008. Till now, he has published two papers in 2010 and 2 in 2011. But he has not been the first author, so there is no record about him in Pubmed, which only displays the organization of first author.

- Lack of organization. Search of Author 'Baciarello G' returns several abstracts, but no organization in it. Normally the organization is displayed after the author list. Following figure clearly shows that there is no organization information.



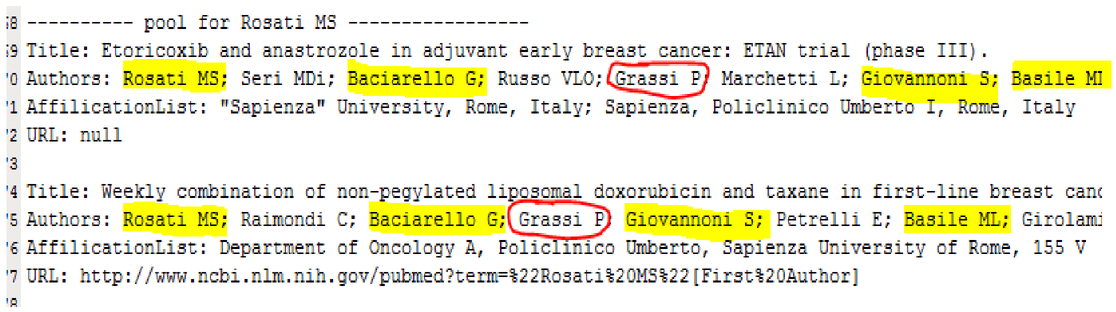
Pacing Clin Electrophysiol. 1983 Mar;6(2 Pt 1):268-71.

High resolution recordings of atrial flutter.

Baciarello G, Di Maio F, Russo GE, Sciacca A.

Abstract

- Organization is in different format and , like author 'Grassi P'. System log shows the matched abstract is found, but it has different organization. The red circled name is the author we are testing; hight-lighten names are the direct co-author's names. It is very apparent that these two abstract are written by the same Grassi P, but the organization is quite different.



```
---------- pool for Rosati MS -----------------
Title: Etoricoxib and anastrozole in adjuvant early breast cancer: ETAN trial (phase III).
Authors: Rosati MS; Seri MDi; Baciarello G; Russo VLO; Grassi P; Marchetti L; Giovannoni S; Basile MI
AffilicationList: "Sapienza" University, Rome, Italy; Sapienza, Policlinico Umberto I, Rome, Italy
URL: null

Title: Weekly combination of non-pegylated liposomal doxorubicin and taxane in first-line breast canc
Authors: Rosati MS; Raimondi C; Baciarello G; Grassi P; Giovannoni S; Petrelli E; Basile ML; Girolami
AffilicationList: Department of Oncology A, Policlinico Umberto, Sapienza University of Rome, 155 V
URL: http://www.ncbi.nlm.nih.gov/pubmed?term=%22Rosati%20MS%22[First%20Author]
```

- No matched abstract is found, like 'Marchetti L'. Pubmed returns 6 abstracts for this author, but the system states they are not the same person. Manual check proves the system is right.

5.5 System Parameter:shrotenInitial

In Chapter 4, we discussed the algorithm of author-disambiguation. To conclude whether or not two abstracts are actually written by the same author, we need to send a web request to check the co-author relationship. To figure out the affect in details, we run the system on the same abstract twice: with or without this parameter set true. Figure 5-6 shows how many web requests are sent with and without setting shortenInitial to be true (the data are list in Appendix B).

Number of Web Page Request Per Author

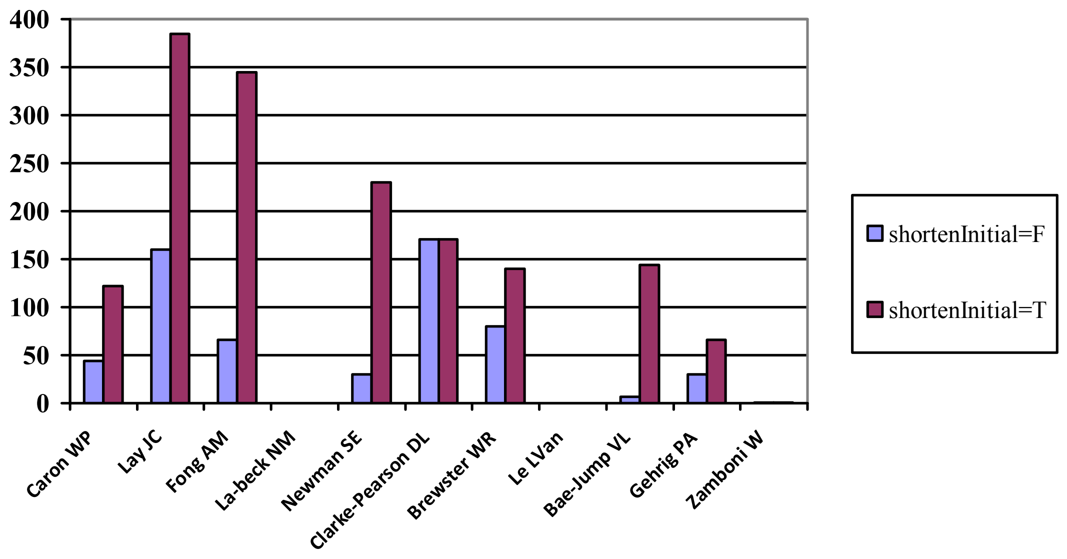


Figure 5-6: Compare chat on web Request for each author

From above figure, it is apparent that when the shortenInitial is true, the system sends more web requests – in fact, the total number of web request sent under shortInitial = false is only 647, but it is 1709 under shortInitial = true. Normally there are more web request when shortenInitial is true, but it may be equal in following situations:

- No records are found for both full name initial and shorten name initial, like the name "La-Beck NM" and "Le LVan"
- There is no middle name, so the full name initial and shorten name initial have the same format, like the name "Zamboni W"
- The full name initial is different from shorten name initial, but no other author has the same shorten name initial, like the name "Clarke-Pearson DL".

In the improved algorithm, database has been added into the system. Before we search the name in the web, we firstly check in the database to see if the non-coauthor relationship has been confirmed. Figure 5-7 Shows how many non-coauthor relationships are confirmed with and without setting shortenInitial to be true. More database confirmation can be got when the shortenInital is true – in fact, the total confirmation is 162 when the parameter is false and it is 562 when false.

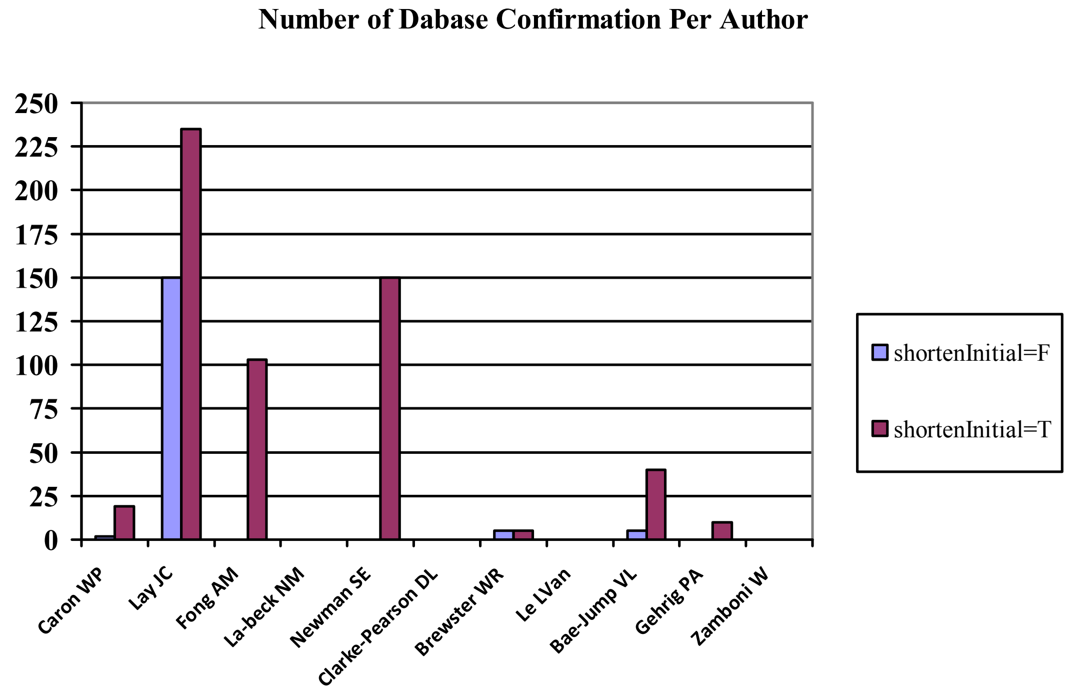


Figure 5-7: Compare chat on how many non-coauthor relationship confirmed in DB for each author under different value of shortenInitial.

5.6 System Parameter: maxMatchedCnt

This parameter decides how many clustered abstract are used to confirm the organization. To get a clustered abstract, massive compare might be needed. So this parameter controls the system speed to some extent. Setting this parameter to be 1 will definitely make the system the fastest, but it might have negative impacts on the correctness: what if the abstract happens to have no organization? Setting it to a big value may fix the potential correctness problem, but it will introduce too many unnecessary compare. To figure out how this parameter will affect the system and to find out what value is the most appropriate, we run the system on the same abstract with different maxMatchCnt.

Figure 5-8 shows how many web requests are sent to disambiguate each author. Though some author has the same web request for all different maxMatchCnt, the trend is quite clear that the larger the maxMatchCnt is, the more web requests will be sent.

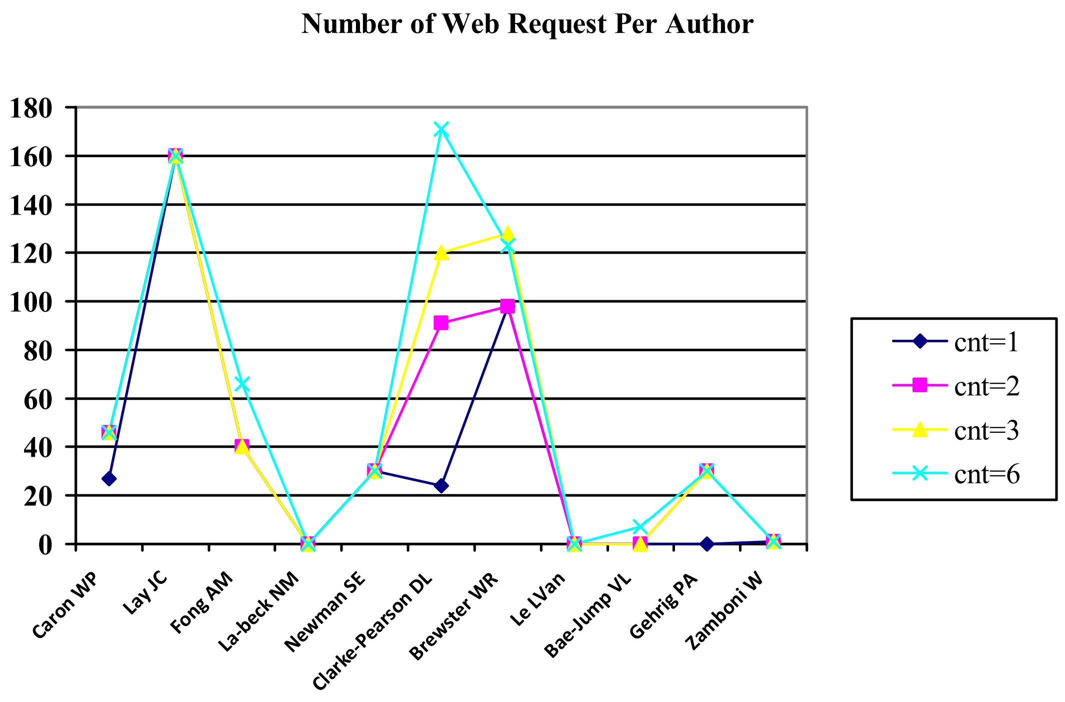


Figure 5-8:Compare chat with different maxMatchCnt.

A typical behaviour should be like the behaviour of author 'Clarke-Pearson DL': different web requests for each macMatchCnt. But the other behaviours are also reasonable and they fall into the following categories:

- No abstracts are found, like 'Le LVan' and 'Zamboni W'. There are tiny difference: 'Le LVan' returns 0 records; while 'Zamboni W' returns several abstracts but none is relevant.
- Only one abstract is found, like 'La-beck NM'. The system can recognize this situation and does not trigger the web search. That is how the 0 got.
- All the abstracts are diagnosed to be written by others. That is, the system checks all the abstracts but finds no matched one, like 'Jay LC'. In total 10 abstracts are returned via the web search, but none is written by the same 'Jay LC' in sample abstract.

Now we can see that a very large maxMatchCnt can force the system to check every single abstract. This will seriously impact the efficiency, but will it disambiguate more author-organization relation? Table 5-3 list the system's performance under different values of parameter maxMatchCnt.

	cnt=1	cnt=2	cnt=3	cnt=6
Total Time (m’s)	8’52	11’56	15’04	15’33
Web Search (times)	399	521	586	678
Non-coauthor DB Comfirmation (times)	222	222	222	232
Author-organization Disambiguation (pairs)	6	7	7	7
Table autho_info (records)	18	24	30	43
Table non_coauthor (records)	375	489	546	622

Table 5-3: System performance with different maxMatchCnt

From the above table, we can see the biggest increment of total time happens from cnt=1 to cnt=2. In this period, the disambiguated author-organization pair also gets increased. The added author-organization is for author ‘Clarke-Pearson DL’ , because the 1st abstract was thought written by different author. This proves that changing parameter maxMatchCnt can impact the correctness.

Above table also shows the increment on the number of web request is not very abrupt, that is because in this abstract, several authors only get a few even zero abstract. For those authors, compare problem will never happen, so the change of maxMatchCnt has little even no impact on them.

The trend of increment will be quite different if authors have many matched abstracts to compare. We take author ‘Clarke-Pearson DL’ for example, because he got the most abstracts, so he will be obviously affected by this parameter.

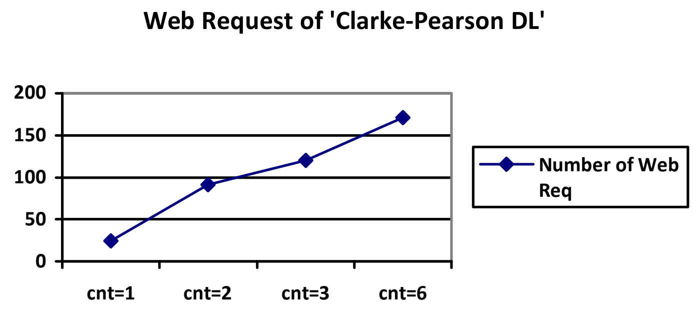


Figure 5-9: Big changes on the number of Web Request

One author will have many publishes in the whole research life, so the parameter maxMatchCnt will probably impact them like what is showed in Figure 5-9.

6. Evaluation

As per customer's requirement, this system should be automated, correct and efficient. In Chapter 3, we mentioned that there are two independent applications in this system: Abstract Extraction module and Author-organization Disambiguation module. In this chapter, we are going to evaluate them.

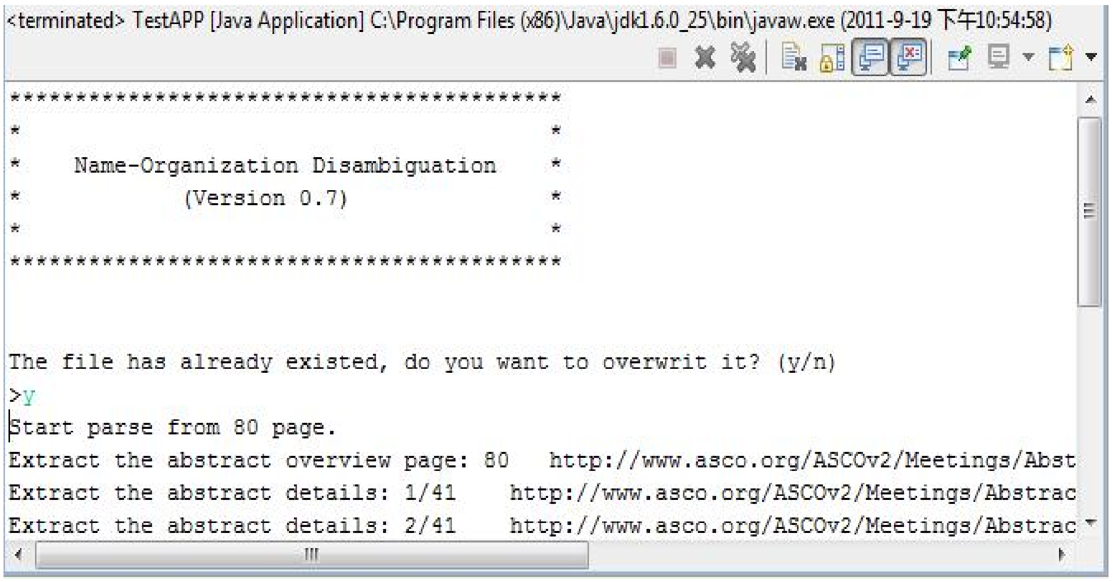
6.1 Abstract Extraction

To evaluate this module, we measure it from the following aspects:

- Correctness. After the extraction, manually compare all the required fields in XML file to that in the web. They should be the same.
- Automated. Given the entry page, check whether the system can find the sub-category page, then find the abstract. We add some debug information into the code to make it output the URL of each extracted abstract. All abstracts should be extracted.
- Efficiency. There is no measurement.

We checked the test result manually and proved the system is correct and automated. As for efficiency, though our customer does not raise the measurable requirements, we think our system is efficient because we designed a special feature called "Continue from Lost". During the extraction, if something happens and interrupts the process because the network gets disconnected, the power is off or even someone stops the program by accident, our system can remember the stopped point and continue from the lost point when it starts again.

We mentioned before that there are in total 80 sub-categories. Each category is converted into an independent XML file. Now we quit the program when it is handling the 80th sub-category and start it again. Figure 6-1 shows the program starts to extract the 80th sub-category directly.



```
<terminated> TestAPP [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_25\bin\javaw.exe (2011-9-19 下午10:54:58)
******************************************
*                                        *
*     Name-Organization Disambiguation   *
*             (Version 0.7)              *
*                                        *
******************************************


The file has already existed, do you want to overwrit it? (y/n)
>y
Start parse from 80 page.
Extract the abstract overview page: 80    http://www.asco.org/ASCOv2/Meetings/Abst
Extract the abstract details: 1/41    http://www.asco.org/ASCOv2/Meetings/Abstrac
Extract the abstract details: 2/41    http://www.asco.org/ASCOv2/Meetings/Abstrac
```

Figure 6-1: Demonstration of "Continue from Lost" Feature.

A fully test run for ASCO Meeting 2011 abstracts all the **4351** abstract and it takes **70** minutes. It is about **1 abstract/second**. Considering we need to find the URL for that abstract, send a request, retrieve the content and convert it into xml

6.2 Author Disambiguation

To evaluate this module, we will compare the disambiguation accuracy to other algorithms. Since in our system, author disambiguation is designed for relationship disambiguation, so we have some minor different evaluation.

6.2.1 Evaluation Criterion

Given an author and one of his abstracts, we expect the algorithm described in Chapter 4 to be able to recognise which are actually the author himself from a collection of identical name identifies. No algorithm can be perfect and our algorithm will not be an exception. However, since author disambiguation is just the first step in the whole disambiguation system, its performance has far-reaching impact. Therefore we have to make sure the algorithm works well.

We define 3 ratios to measure how our "co-author based author disambiguation algorithm" performs.

- **Ratio of Correctness (RoC):** Given an author, if our system deduces an identical name is the author or if our system thinks of it as a different person and it is actually true, we call it a 'correct cluster'. RoC can be calculated in Formula 6-1:

  $$\text{RoC} = \frac{(\text{The number of correct cluster})}{(\text{The number of clusterred names})} \times 100\% \qquad \text{(formula 6-1)}$$

  RoC indicates our system's abilities on author disambiguation. The higher the RoC is, the better performance it is. We expect the RoC will be about 75%.

- **Ratio of Missing (RoM):** Given an author, if our system deduces that one name is just an identical name but in fact it is referring to the same person, we call it a 'missing cluster'. RoM can be calculated in Formula 6-2:

  $$\text{RoM} = \frac{(\text{The number ofmissing cluster})}{(\text{The number of clusterred names})} \times 100\% \qquad \text{(formula 6-2)}$$

  For a system that requires a high accuracy of author disambiguation, RoM should be very low. Our system is a little tolerant on this, because it cares about correctness more. So in our system RoM is allowed to be as high as 20%.

- **Ratio of Incorrectness (RoI):):** Given an author, if our system deduces that one name is referring to the same person but in fact it is just an identical name, we call it an 'incorrect cluster'. RoI can be calculated in Formula 6-3:

$$\text{RoI} = \frac{\text{(The number of incorrect cluster)}}{\text{(The number of clusterred names)}} \times 100\% \qquad \text{(formula 6-3)}$$

RoI indicates how far our system is being wrong. An ideal system's RoI should be equal to 0. As we mentioned before, author-organization disambiguation is built on author disambiguation. An 'incorrect cluster' can deduce a wrong organization or a nil organization. We try to keep our system's RoI as low as we can and will not allow it to be above 5%.

6.2.2 Evaluation

According to test data in Table 5-2, the name disambiguation performance quit good. The RoC = 71/81= 87.66%. Compare with other algorithm [16, 17], our performance is good. The compare result is in Table 6-1.

	Our System	K-way on DBLP	DBSCAN
Accuracy	87.66%	62.03%	90.6%

Table 6-1: Disambiguation accuracy compare table

The reason why our algorithm has such high accuracy is because we run this program on a very small-scale data. In the large-scale data, the accuracy will decrease, at about 70%. It is good because we only make use of the co-authors.

The enhanced algorithm can improve the efficiency by doing the necessary compare only. To evaluate how well it works, we run both origin algorithm and new algorithm for the same abstract. The system will record the start time, end time, number of web request, number of non-coauthor relationship found in database.

We put both test result in Figure 6-2, it is obviously the enhanced algorithm is more efficient.

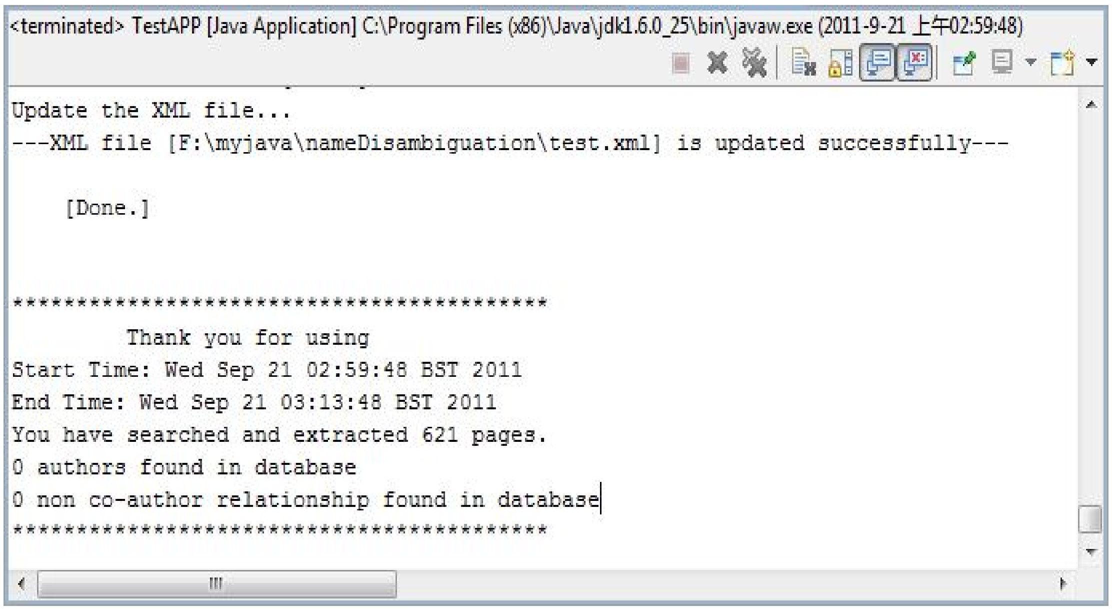
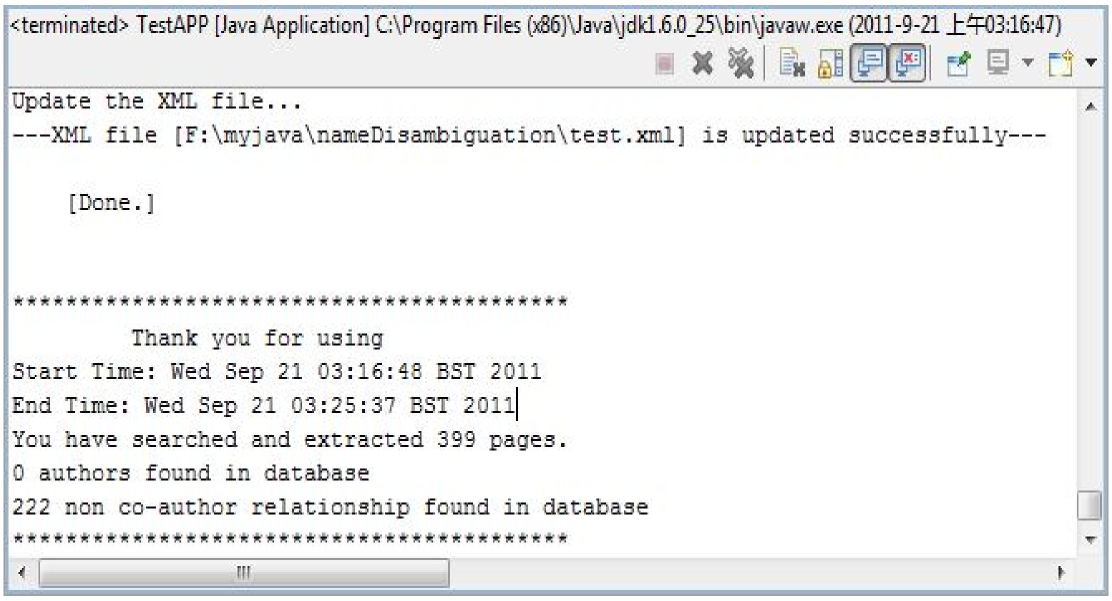

```
<terminated> TestAPP [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_25\bin\javaw.exe (2011-9-21 上午02:59:48)
Update the XML file...
---XML file [F:\myjava\nameDisambiguation\test.xml] is updated successfully---

    [Done.]


*******************************************
         Thank you for using
Start Time: Wed Sep 21 02:59:48 BST 2011
End Time: Wed Sep 21 03:13:48 BST 2011
You have searched and extracted 621 pages.
0 authors found in database
0 non co-author relationship found in database
*******************************************
```

Figure 6-2 (a): Origin algorithm spent 14 minutes and did the web search 621 times.

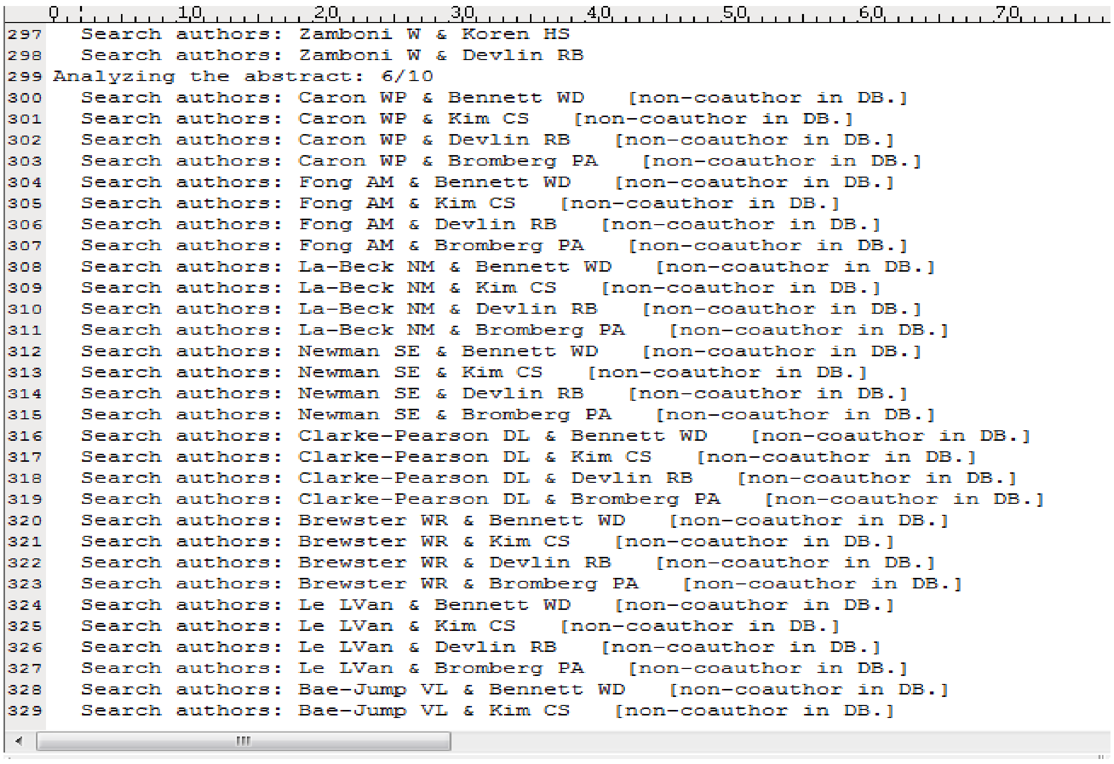


<terminated> TestAPP [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_25\bin\javaw.exe (2011-9-21 上午03:16:47)

```
Update the XML file...
---XML file [F:\myjava\nameDisambiguation\test.xml] is updated successfully---

    [Done.]


******************************************
         Thank you for using
Start Time: Wed Sep 21 03:16:48 BST 2011
End Time: Wed Sep 21 03:25:37 BST 2011
You have searched and extracted 399 pages.
0 authors found in database
222 non co-author relationship found in database
******************************************
```

Figure 6-2 (b): Enhance algorithm spent only 9 minutes and did the web search 399 times.

To prove the database indeed can reduce the web request, we make a minor change in the system. If one pair of authors is found in table NON_COAUTHOR, a simple prompt message '[non-coauthor in DB.]' will be printed. In Figure 6-3, we can see most of relationship in the 6th abstract is confirmed without requesting the web.



```
  Search authors: Zamboni W & Koren HS
  Search authors: Zamboni W & Devlin RB
Analyzing the abstract: 6/10
  Search authors: Caron WP & Bennett WD   [non-coauthor in DB.]
  Search authors: Caron WP & Kim CS   [non-coauthor in DB.]
  Search authors: Caron WP & Devlin RB   [non-coauthor in DB.]
  Search authors: Caron WP & Bromberg PA   [non-coauthor in DB.]
  Search authors: Fong AM & Bennett WD   [non-coauthor in DB.]
  Search authors: Fong AM & Kim CS   [non-coauthor in DB.]
  Search authors: Fong AM & Devlin RB   [non-coauthor in DB.]
  Search authors: Fong AM & Bromberg PA   [non-coauthor in DB.]
  Search authors: La-Beck NM & Bennett WD   [non-coauthor in DB.]
  Search authors: La-Beck NM & Kim CS   [non-coauthor in DB.]
  Search authors: La-Beck NM & Devlin RB   [non-coauthor in DB.]
  Search authors: La-Beck NM & Bromberg PA   [non-coauthor in DB.]
  Search authors: Newman SE & Bennett WD   [non-coauthor in DB.]
  Search authors: Newman SE & Kim CS   [non-coauthor in DB.]
  Search authors: Newman SE & Devlin RB   [non-coauthor in DB.]
  Search authors: Newman SE & Bromberg PA   [non-coauthor in DB.]
  Search authors: Clarke-Pearson DL & Bennett WD   [non-coauthor in DB.]
  Search authors: Clarke-Pearson DL & Kim CS   [non-coauthor in DB.]
  Search authors: Clarke-Pearson DL & Devlin RB   [non-coauthor in DB.]
  Search authors: Clarke-Pearson DL & Bromberg PA   [non-coauthor in DB.]
  Search authors: Brewster WR & Bennett WD   [non-coauthor in DB.]
  Search authors: Brewster WR & Kim CS   [non-coauthor in DB.]
  Search authors: Brewster WR & Devlin RB   [non-coauthor in DB.]
  Search authors: Brewster WR & Bromberg PA   [non-coauthor in DB.]
  Search authors: Le LVan & Bennett WD   [non-coauthor in DB.]
  Search authors: Le LVan & Kim CS   [non-coauthor in DB.]
  Search authors: Le LVan & Devlin RB   [non-coauthor in DB.]
  Search authors: Le LVan & Bromberg PA   [non-coauthor in DB.]
  Search authors: Bae-Jump VL & Bennett WD   [non-coauthor in DB.]
  Search authors: Bae-Jump VL & Kim CS   [non-coauthor in DB.]
```

Figure 6-3: 'non-coauthor in DB' in output window

6.3 Relationship Disambiguation

We have separately introduced the author disambiguation algorithm and the organization confirmation algorithm in prior chapters. Based on the performance, the whole system is decided to adopt Co-author based Author Disambiguation along with Sole Information Compare in order to finish the author-organization disambiguation.

Since there is no relationship disambiguation algorithm so far, we can only evaluate it with test data. Considering the fact that no enough organization information can be retrieved and our algorithm is experimental, we think we achieve the goals as long as the whole accuracy is about 50%.

We run the system on parts of the sample abstracts, which have clear author-organization relationship. Then use the file compare tool to find any different between the sample abstract and our system's output file. The test Result is list in Table 6-2, as below.

Total Number of sample authors:	805
Total Number of Successful disambiguation:	434 434 / 805 = **53.91**%
Total Number of Incorrect disambiguation:	0
Total Records Saved into Non-Coauthor Table:	58001
Total Records Saved into Coauthor Table:	1550

Table 6-2: System Performance on large scale data

Each algorithm has already been proved of good performance. However, when they come together, it is not a simple plus. If we have a great deal of information, there is no surprise that we can reduce the correct author-organization relationship. But massive information in adverse will decrease the efficiency in two aspects:

a) Irrelevant information is everywhere and there are more in massive information. Even the ratio is the same, let us assume it 10%, the number of irrelevant information is much larger. If there are 10 information, the number of irrelevant information is 10 x 10% = 1; now if we have 100 information, the number increases to 100 x 10% = 10. Irrelevant information never shows itself, so we have to spend time to check it and recognize it. More information, therefore, means more time has to be spent on it.

b) Processing relevant information also needs time. A simple but not accuracy formula of calculating the number can be describe as: $T_{total\_time} = T_{unit\_time} \times N_{information}$ From this formula, we can see that T_{total_time} is proportional to $N_{information.}$ The increment of information number must result in the increment of process time.

As all the other systems, our system also has to comprise and keeps balance between efficiency and correctness. There are several factors which have big impact on the system's performance. We will experiment them one by one and find out what and how they can influence the system.

6.4 shortenInitial

Though more relationship can be confirmed by the database which helps to reduce the number of web request, the total execution time probably can not be reduced impressively. This is because database operation happens on the hard disk. With the modern hardware technology, I/O operation on disk is very fast. A single search in database only takes 1 or 2 ms. However, Sending web request and receiving the web content are time-consuming, especially when the network is not very good.

By comparing the system performance listed in Table 6-3, we can know the following facts:

- The total processing time is proportional to the number of web request, the function can be described as $T_{total\_time} = a\ N_{web\_request}$ ($1.33 < a < 1.35$).
- The number of disambiguated author-organization relationship is the same. In other words, this parameter can not impact the correctness. It is true in this speculate case, because Pubmed does not mess up the middle name. For name 'Caron WP', it always display as 'Caron WP'.

	Total Time	**Disambiguated Pair**	**Web Request**	**Dabatase OP**	**Avarage**
shortenInitial=F	14m30s	7/11	647	162	1.345 s/request
shortenInitial=T	38m5s	7/11	1709	562	1.337 s/request

Table 6-3 : System performance

In a word, parameter shortenInitial can enlarge the search range and get more results. More information in theory can improve the correctness of disambiguation, but decrease the efficiency. Our system is a practical application against specified websites (Pubmed here), which always displays the middle name if it exists, so this parameter is unfortunately useless in our system.

7. Future Extension and Conclusion

As a prototype system, our system performs well. It has designed and implemented the core algorithm; it introduces the configuration file and makes the system more flexible; it provides the 'Continue from where lost' feature to avoid waste of time and efforts. Though we made some progress, we have to admit that we are still a little far away from turning the system into an official one.

7.1 Author-disambiguation algorithm

The disadvantage of co-author based author-disambiguation algorithm is that it required too many times of web search, which potentially will increase the execution time. No optimization is suitable because the speed of web search mainly depends on the network. If the band width is very narrow, the searching time will be longer; if the web page contains pictures, audio files or video files, the size of content becomes large so it will also take more time; the worst thing is the network is not stable and may be disconnected from time to time. Even we have designed a feature called ‘continue from lost’. It can guarantee there is no loss at all. After all, this feature is based on the abstract. If we have 10 authors and the connection gets disconnected when the system is process the 5th author. When the system goes back to the service, it starts from the 1st author again.

In the future, we can try to add weight into the co-author disambiguation. Currently each co-author relationship has the same weights, so publishing with an author only once and publishing with the other several times have no difference in our system. If we want to introduce the weight, we can consider the frequency of two authors’ co-author relationship. In the graph, we still use the node to present one author, the connection between two nodes means these two authors have co-author relationship and the weight describes how many times they have published together, as shown in Figure 7-1.

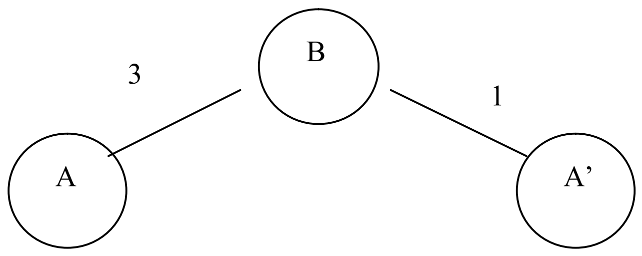


Figure 7-1: Co-author relationship with weights.

After introducing the weights, we only count the direct co-author relationship. For a researcher, the person who has the most possibilities to publish with him several times is one who does the researches with him. So we can set a threshold value and believe two nodes have the same organization if their weights are larger than the threshold value. If the threshold is set to be 2, then the system thinks A and Author, whose weight is 3, works in the same organization. In this case, once we confirm A’s organization, we can get B’s organization at the same time.

Weights can also used by indirect co-author relationship. At the current stage, our system only admits the first level indirect co-author relationship. If weights are applied, we can consider more. If we have 3 identical names A, A’ and A’’ in a co-author relationship like in Figure 7-2, current algorithm will treat them as three individuals. We can see that A,B,C,D and A’ have higher weights, so in this case we can think A and A’ are the same author. For A’’, he will be thought as different because its weight is only 1.

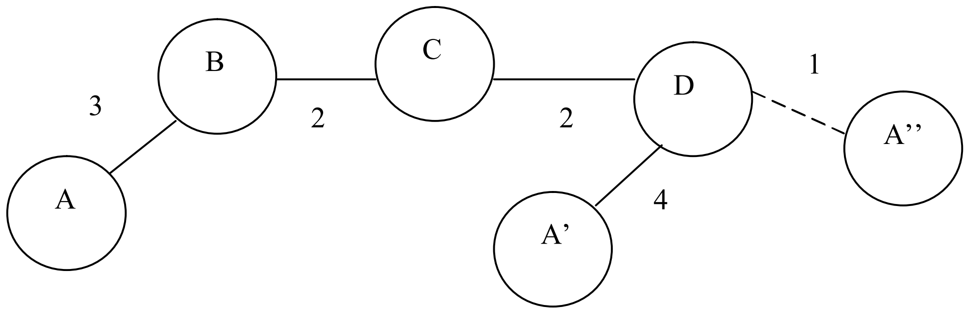


Figure 7-2: Indirect co-author relationship with weights

With the use of weights, the system's performance should be optimistically better. Above two figures indicates it may help to reduce the organization directly for those highly connected nodes; it may also increases our system's abilities to do the indirect co-author relationship check. However, for those new researchers who just have few works so far, the weights are not very helpful either.

7.2 Organization-confirmation Algorithm

Another part which needs to be improved in the future is the organization-confirmation algorithm. Our system has already been able to do the text compare with its context, but it is only at the entry level. In the future, we need to find a better and flexible way.

So far, only two or three different formats can be recognized by our system. To improve the service, we can do a research on the organization format used by different web pages. Then we can summarize them and build a list of format – plain text file will be enough. Every time we do an organization to confirmation check, we can go though the whole list. Once it turns out to be the same, we end the problem and think they are the same.

7.3 Consider the time factor

A researcher usually changes his organization during his research work: at the beginning he did research in his university, then he would find a research job in a company after the graduation, then he maybe change his research field, therefore change his organization. So for one author, we can get more than one organization. Can we take advantage of the information? The answer is yes.

The idea is to record the year when a different organization is found for the same author. Changing organization is common but frequent change is seldom. So we may get the organization information according to the year information. Figure 7-3 shows an organization-year coordinate, that author worked in Organization C from 1998 to 2004, worked in Organiztion A from 2004 to 2007 and worked in Organization B since 2007. If now there is another paper written in 2005, it is very possible that author was in Organization B when he wrote the paper.

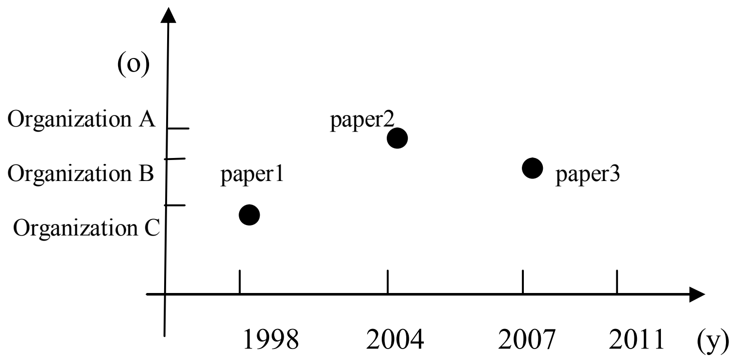


Figure 7-3: The organization-year coordinate for an author

This solution is simple and reasonable, but confusion in some cases. One of the most important questions is what is the safe zone if we get an organization for a specific time? In

Figure 7-3, we get Organiztion A of 2004, can we sure the author did not change his organization during 2004 to 2007? If we can not, can we make a reasonable assumption on that? Even we know the average stable time, this solution still can result incorrect answer because average rule can not apply to everyone.

7.4 Conclusion

Author disambiguation is never an easy task and it becomes more difficult with the consideration about organization. Most research contributes to the single entity disambiguation, so our system is a fresh experiment in relationship disambiguation.

Inspired by the Heuristics Graph used in digital library area, we proposed Two Stage algorithm. In this algorithm, the whole relationship disambiguation is split into two stages: the first stage is to collect the information of that author by using Co-author Based Author Disambiguation algorithm; the second stage is to reduce the organization information, according to the data collected in last stage.

Test proves the good performance of each stage, but the overall relationship disambiguation ratio is only about 53.91%. Analysis of the failure results reveals that lack of enough information is the main reason. Another reason is the limitation of Organization Confirmation algorithm. If two organizations are too different in the format, the system may fail to recognize them.

Though the disambiguation ratio is a little low, as a prototype system, we think our system has tried its best. We introduce the database and save all the middle results, to reduce the need of sending web request. By doing this, the system's efficiency can increase twice. With more and more information are saved into the database, our system will run faster and faster. We also have tried to find information via search engine, but the result is not promising. We did what we can, but the performance is not improved a lot because the system does not have enough information to reduce the relationship. However, we propose some ideas at the end of this dissertation, which may be helpful if anyone is interested in the enhancement of this system in the near future.

Abstract extraction module is an assistant module for the relationship disambiguation. Our system performs perfectly in this part. The system has 100% correctness on the extraction. Continue from Lost feature guarantees the system can continue the work from the nearest point away from where it stopped last time.

In a word, as a practical system, our system successfully keeps balance between correctness and efficiency.

Bibliography

1. Tan, Y.F., Kan, M.-Y. and Lee, D. *Search Engine Driven Author Disambiguation*, Proceedings 6th ACM/IEEE-CS Joint Conference on Digital Libraries, pp.314-315, ACM Press, New York

2. I. P. Fellegi and A. B. Sunter. A *theory for record linkage*. Journal of the American Statistical Association, 64:1183–1210, 1969.

3. L. Gu, R. Baxter, D. Vickers, and C. Rainsford. *Record linkage: Current practice and future directions*. In CMIS Technical Report No. 03/83, 2003.

4. *Overview - Web of Science*, Thomson Reuters. 2011. Retrieved 2011-05-11

5. M. Uschold and M. Gr¨uniger. *Ontologies: Principles, methods and applications*. Knowledge Engineering Review, 11(2):93–155, 1996.

6. Chung Heong Gooi and James Allan. 2004. *Cross-document coreference on a large scale corpus*. In Proceedings of Human Language Technology Conference / North American Association for Computational Linguistics Annual Meeting, Boston, MA.

7. Bagga and B. Baldwin. 1998. *Entity-based cross-document coreferencing using the vector space model*. In Christian Boitet and Pete Whitelock, editors, Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics, pages 79–85, San Francisco, California. Morgan Kaufmann Publishers.

8. Elmagarmid, A., Ipeirotis, P., Verykios, V. (2007). *Duplicate Record Detection: A survey*. IEEE Transactions on Knowledge and Data Engineering 19(1):1–16.

9. Dror G. Feitelson. *On identifying name equivalences in digital libraries*. Inf. Res, 9(4), 2004.

10. Dongwon Lee, Byung-Won On, Jaewoo Kang, and Sanghyun Park. Effective and scalable solutions for mixed and split citation problems in digital libraries. In IQIS ’05: Proceedings of the 2nd international workshop on Information quality in information systems, pages 69–76, New York, NY, USA, 2005. ACM Press.

11. Luis Gravano, Panagiotis G. Ipeirotis, Nick Koudas, and Divesh Srivastava. *Text joins in an RDBMS for web data integration*. In WWW ’03: Proceedings of the 12th international conference on World Wide Web, pages 90–101, New York, NY, USA, 2003. ACM Press.

12. C. Frankel, M. Swain, and V. Athitsos. *Webseer: An image search engine for the world wide web*. Technical Report 96-14, University of Chicago, August 1996.

13. Danny Sullivan, ed., *How Search Engines Work*, Search Engine Watch, <http://www.searchenginewatch.com/webmasters/work.html>, accessed March 14, 1999

14. *Scopus in detail: What does it cover?*. Scopus Info. Elsevier. Retrieved 2011-05-11.

15. *Overview of SpingerLink*, < http://www.springerlink.com/home/main.mpx>, Springer. Retrieved 2011-05-11.

16. Han, H., Zha, H., Giles, C.L.: *Name disambiguation in author citations using a K-way spectral clustering method*. In: Proceedings of JCDL. (2005) 334–343

17. J. Huang, S. Ertekin, and C. L. Giles. Efficient name disambiguation for large-scale databases. In *the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 536–544. Springer-Verlag Berlin Heidelberg, 2006.

Appendix A: Test Data

Database	Author	Results	Author	Results	Author	Results	Author	Results
CiteSeer	Caron WP	0	Lay JC	0	Fong AM	0	La-Beck NM	0
DBLP	Caron WP	0	Lay JC	0	Fong AM	2	La-Beck NM	0
MedlinePlus	Caron WP	0	Lay JC	5	Fong AM	0	La-Beck NM	0
pubMed	Caron WP	2	Lay JC	44	Fong AM	28	La-Beck NM	2

Table A-1: Search Result of Author in Abstract of Figure 1-1 (1)

Database	Author	Results	Author	Results	Author	Results	Author	Results
CiteSeer	Newman SE	0	Brewster WR	0	Le LV	0	Gehrig PA	0
DBLP	Newman SE	0	Brewster WR	0	Le LV	2	Gehrig PA	0
MedlinePlus	Newman SE	2	Brewster WR	0	Le LV	0	Gehrig PA	14
pubMed	Newman SE	28	Brewster WR	50	Le LV	1	Gehrig PA	55

Table A-1: Search Result of Author in Abstract of Figure 1-1 (2)

Database	Author	Results	Author	Results	Author	Results
CiteSeer	Bae-Jump VL	0	Zamboni W	0	Clarke-Pearson DL	0
DBLP	Bae-Jump VL	0	Zamboni W	0	Clarke-Pearson DL	0
MedlinePlus	Bae-Jump VL	0	Zamboni W	0	Clarke-Pearson DL	0
pubMed	Bae-Jump VL	12	Zamboni W	7	Clarke-Pearson DL	157

Table A-1: Search Result of Author in Abstract of Figure 1-1 (3)

Appendix B: Test Data of System Performance

Author	Abstract Found	Web Req	DB	Abstract Found	Web Req	DB
W.P. Caron	2	44	2	8	122	19
J.C. Lay	10	160	150	15	385	235
A.M. Fong	6	66	0	19	345	103
N.M La-Beck	0	0	0	0	0	0
S. E. Newman	2	30	0	16	230	150
D. L. Clarke-Pearson	16	171	0	16	171	0
W. R. Brewster	6	80	5	7	140	5
L. Van Le	0	0	0	17	0	0
V. L. Bae-Jump	5	7	5	6	144	40
P. A. Gehrig	11	30	0	16	66	10
W. Zamboni	18	1	0	18	1	0