Journal of Universal Computer Science, vol. 15, no. 8 (2009), 1685-1710
submitted: 22/11/08, accepted: 24/4/09, appeared: 28/4/09 © J.UCS

Discovering Semantic Aspects of Socially Constructed Knowledge Hierarchy to Boost the Relevance of Web Searching

Dengya Zhu
(Curtin University of Technology, Perth, Australia
dengya.zhu@postgrad.curtin.edu.au)

Heinz Dreher
(Curtin University of Technology, Perth, Australia
h.dreher@curtin.edu.au)

Abstract: The research intends to boost the relevance of Web search results by classifying *Websnippet* into socially constructed hierarchical search concepts, such as the most comprehensive human edited knowledge structure, the Open Directory Project (ODP). The semantic aspects of the search concepts (categories) in the socially constructed hierarchical knowledge repositories are extracted from the associated textual information contributed by societies. The textual information is explored and analyzed to construct a *category-document* set, which is subsequently employed to represent the semantics of the socially constructed search concepts. Simple API for XML (SAX), a component of JAXP (Java API for XML Processing) is utilized to read in and analyze the two RDF format ODP data files, *structure.rdf* and *content.rdf*. kNN, which is trained by the constructed *category-document* set, is used to categorized the Web search results. The categorized Web search results are then ontologically filtered based on the interactions of Web information seekers. Initial experimental results demonstrate that the proposed approach can improve *precision* by 23.5%.

Keywords: Web search, semantic analysis, ontology, the Open Directory Project, socially constructed knowledge repository, SAX, HTML
Categories: H.3.3, H.3.4

1 Introduction

The advent and the dramatic growth of the Web impose many challenges to information retrieval. As the dominant Web information retrieval tool, due to information explosion on the Web, search engines are also facing a number of issues [Zhu and Dreher 2007], such as 1) the polysemous and synonymous characteristics of natural languages; 2) information overload; and 3) the gap between information needs represented by search-terms and the ranked search result list provided by information retrieval systems which estimate the relevance of the search results relative to the search-terms.

Leveraging external domain knowledge is a promising approach to boost information retrieval [Turtle and Croft 1996]. Text categorization, also called classification, or supervised learning [Sebastiani 2002], is one of the technologies which can be used to automatically assign predefined categories to free text

documents [Yang 1999]. It has been applied in the field ranging from document indexing, filtering, automated metadata generation, and Web resource organization [Sebastiani 2002].

For text categorization, one issue that needs to be addressed is the expense of obtaining training data which is essential for learning a categorization algorithm - training data creation involves human labour.

Another concern of text categorization is the predefined category set into which free documents are classified. For example, some researchers employ Yahoo! Web Directory as the predefined category [Labrou and Finin 1999, Mladenic 1998], others use the Open Directory Project [The Open Directory Project 2008] as the Web directory [Chirita, et al. 2005, Gauch, et al. 2003, Zhu and Dreher 2007]; the Reuters Collection is widely utilized for research purposes [Deng, et al. 2004, Lewis, et al. 2004, Yang 1999, Yang and Liu 1999].

For each of the categories in the ODP, the most comprehensive, human edited, socially constructed Web knowledge hierarchy, there is a list of Web sites each of which has a title and a brief description. This means that the content of these Web sites are relevant to the corresponding category. There are professional volunteer human editors to ensure the submitted Web sites are classified under appropriate categories.

This research concentrates on exploring and discovering the data from socially constructed hierarchical knowledge repositories such as the ODP, to represent the semantics of the concepts (categories) in the hierarchical knowledge structure. The proposed approach is to use the Simple API for XML to extract and analyze the data from the two ODP data files: the *structure.rdf* and the *content.rdf*; and utilize these unstructured data to create a textual *category-document* set. Each of the ODP categories has a corresponding *category-document* representing the semantic characteristics of the category. The *category-document* set can then be used as training data, and kNN algorithm [Mitchell 1997] can be used to classify a set of *Websnippet*, or any retrieved information from an information retrieval system. We define a *Websnippet* as an item in the list of Web search results returned by search engines. A *Websnippet* usually contains only the title of a Web page and an optional very short (less than 30 words) description of the page [Zhu and Dreher 2008].

The above three issues can be addressed by text categorization using the socially constructed hierarchical knowledge structure. “Jaguar” is an ambiguous word; “panthera onca” and “jaguar” are synonymous. Searching for “jaguar” by search engines will result in a flat list of tens of millions of Web search results about the animal jaguar, jaguar car, operating system Jaguar, jaguar flight, and anything literally related to the word jaguar. “Washington” is another ambiguous word that may refer to the Capital of US, the first President of US, a boxing trainer, or the federal government of US; “Ford”, another example, may refer to Ford Motor Company, the 38th President of US, or a verb that means cross a river. By comparing the semantics of the categories in the knowledge repository with each of the returned *Websnippet*, the snippets can be arranged into the categories of the knowledge repository according to the calculated semantic similarities and thus alleviate the polysemy issue. At the same time, the synonym problem can also be addressed because the search results relevant to jaguar and panthera onca will have the same latent semantic structure [Deerwester, et al. 1990] and will be grouped into the same category. When

the categorized results are presented to a user, and an interesting category is selected, only the results classified under the selected category are presented, other results are filtered out. If the selected search results still contain ambiguous information, a further refined results set will be presented to the user based on the user's interaction. This will greatly reduce the number of results presented to the user, and thus alleviate the information overload issue. In this interactive Web searching process, the user himself/herself points out which category is of interest, and disambiguation is actually achieved by the user personally. This leads to the proposition of a gyroidal pyramid interactive Web information retrieval model, as shown in Figure 1.

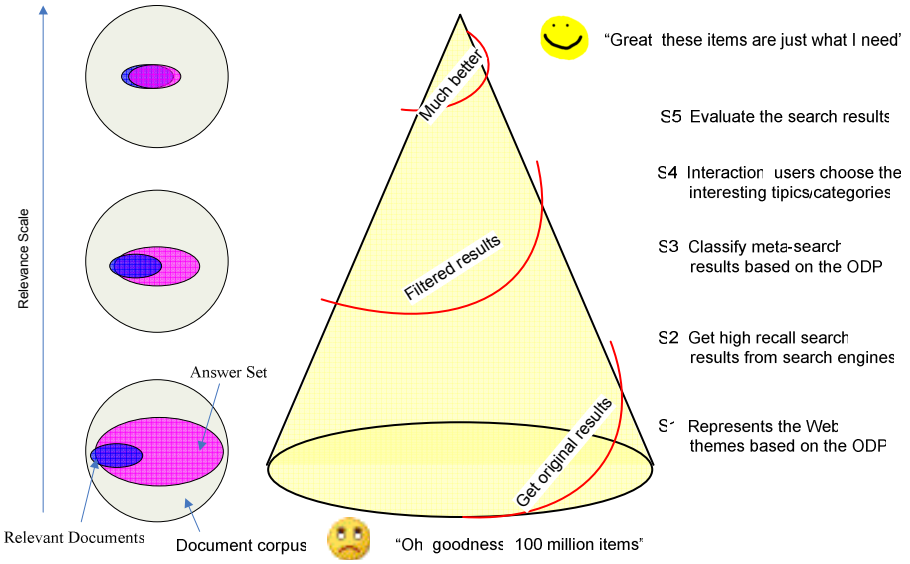


Figure 1: Gyroidal Web searching procedure [Zhu 2007]

In this interactive information retrieval procedure, a user first has a general information need in mind, for example, try to find some information about the animal jaguar. The user selects a search engine - for example, Google - and uses jaguar as a search-term to perform a search. Google will respond to the query, and return millions of search items. Of the top ten ranked search results presented in the first page, perhaps only one or two are relevant to the animal jaguar; most of the search results are irrelevant. The user has to select amongst the dispersed and distributed relevant results from the long list of the returned *Websnippet* objects page by page.

In our proposed information retrieval system, the Web search results will be categorized according to a socially constructed knowledge hierarchy, such as the ODP. The user is allowed to select an interesting category, and only the results classified under the category are presented. These results are further categorized into the next level of the knowledge hierarchy, and the user can obtain refined results if a more specific category is selected. This process can continue until the user is satisfied with the search results. This process is illustrated in Figure 1, step S1 to step S5. In

step S1, the Web is represented by a tree-like Web directory, or a predefined knowledge structure, such as the ODP. The user submits search-terms to a search engine and obtains a list of search results, perhaps most of them are irrelevant (step S2). Search users would feel dissatisfied and overwhelmed – represented by the disappointed user at the bottom of Figure 1. The long list of *Websnippets* is categorized into the predefined knowledge structure in step S3. The user can then choose an interesting topic/category in the knowledge structure (step S4), and evaluate the refined search results classified under the selected topic/category (step S5). The user can choose another topic/category to obtain another set of search results; or select a more specific topic/category in the knowledge hierarchy. This process can be repeated until the user's information need is satisfied – depicted at the top in Figure 1 by the similing user.

2 The Open Directory Project Structure

The ODP was set up by Skreta and Truel in June 1998 in response to the shortcomings of Yahoo! Web Directory [Sherman 2000]. Maintained by a small group of editors, the growth of Yahoo! directory could not keep pace with the explosive growth of the Web. Spurred by the success of Open Source movement, the ODP originators reasoned that a Web directory could keep up with the dynamic nature and rate of change of the Internet, if there were enough volunteer editors to index the Internet. Practice has proven they are correct. Since the creation of the ODP, the number of volunteers, the indexed Web pages, and the categories are all growing rapidly [Sherman 2000].

Today, the ODP is the largest, most comprehensive human-edited directory of the Web [The Open Directory Project 2008]. It now contains over 4.59 million submitted Web sites, 82,929 editors and 590,000 categories, and these numbers are increasing continuously. The size of the RDF/XML content file in gz compressed format of the ODP is now 302MB; the structure file in gz compressed format is 72MB (Accessed on April 7, 2009).

The reason we chose the ODP repository in our research is that, first, the knowledge structure of the ODP can be taken as our pre-defined categories for *Websnippet* classification because the ODP category is dynamic which is trying to keep pace with the explosive growth of the Web, and thus very suitable for *Websnippet* categorization; second, it is organized by human editors and consequently more authority than automatically produced knowledge hierarchies, such as those constructed by clustering algorithms [Jain, et al. 1999]; third, as the most comprehensive human edited knowledge hierarchy, the ODP also provides millions of information items which manifest the semantics of the categories by the informative, descriptive, and concise features of the information items. These huge amounts of information are subsequently employed as training data sets. The following two sections present a detailed discussion on how the ODP data is analyzed and extracted to manifest the semantic characteristics of the ODP categories.

2.1 The ODP

The ODP is a Web directory of Internet resources and it is the most widely distributed data base of Web content classified by humans [The Open Directory Project 2008]. A Web directory is somewhat like a huge reference library. The directory is arranged in a hierarchy, the broader topic is on the higher level of the structure, the more specific subject is placed in the lower level of the structure. All the Web pages submitted to the ODP are subject to human editor evaluation [The Open Directory Project 2008].

Each of the categories in the ODP contains the *title* and the *topic* of the category, a number of *subcategories,* a *description* of the category, and a list of *submitted Web pages*. The topic of the category is specified using a concatenation of categotry names from the root (or TOP) to the category (or topic) of focus. For example, "Top: Science: Biology: Flora and Fauna" is the topic of the category "Flora and Fauna". The title of a category does not include its super-categories; "Flora and Fauna" is the title of category with the topic "Top: Science: Biology: Flora and Fauna". The *description* of the category is usually a further explanation of the meaning of the category, and some information about the content and subject matter [The Open Directory Project 2008].

Some categories may also contain editorial information to emphasise what kind of Web sites should not be submitted under this category. This editorial information is not semantically related to the category, and will not be extracted as the semantic characteristics of the category. For each submitted Web page, beside the *title* to identify the site, there is also a concise and accurate *description* of the Web page which tells the end users what they will find when the site is visited.

2.2 The Hierarchical Structure of the ODP

Categories in the ODP are hierarchically structured as shown in Figure 2. From the root category, the ODP (or TOP), there are 15 first level categories. In addition to the 15 categories, category "World" supports the ODP in different languages. Figure 3 shows the 15 + 1 categories.

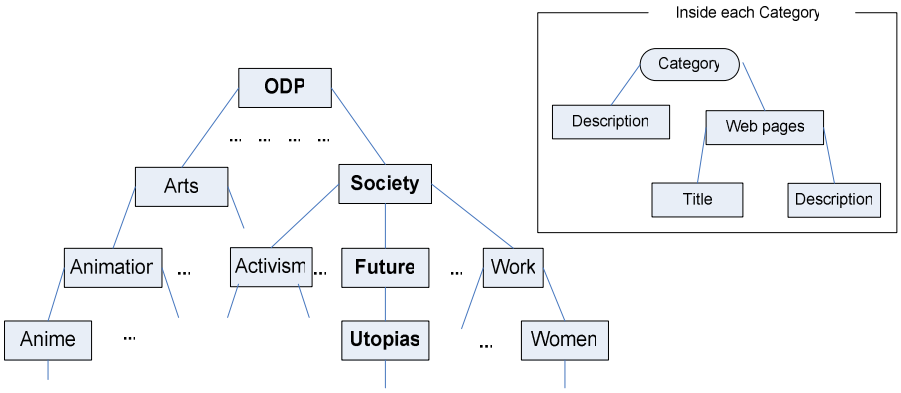


Figure 2: The Hierarchical Structure of the ODP [Zhu 2007]

Each of the fifteen first level ODP categories has its own subcategories. For example, under category "Society", the level two subcategories are: Activism; …; Future; …; and Work. Figure 4 is a screenshot of the subcategories of the category "Society". Subcategories under the category "Society" may have their own sub-subcategories, and these sub-subcategories may in turn each have their own subcategories, until a category reaches the end (leaf node) of the tree structure. The deepest level of the ODP is 14, with an average level of 10.65 [Perugini 2008].

Each category in the ODP can be identified by the topic of the category. For example, "Top: Society: Future: Utopias" is the topic of the category titled "Utopias"; its direct supercategory is "Future"; its first level supercategory (immediately after "Top") is "Society".

Figure 3: The 15+1 First Level Categories of the ODP (www.dmoz.org)

One subcategory may be arranged under more than one category in the ODP. If an @ character runs after the name of a subcategory, it indicates that although this subcategory is arranged under this category, it is originally categorized at another category. For example, in Figure 4, there is an @ character running after the category "Economics". When "Economics" is clicked, it reaches "Top: Science: Social Sciences: Economics". This indicates that the category "Economics" is originally categorized under the category "Top: Science: Social Sciences: Economics"; nevertheless, it can also be classified under "Top: Society".

2.3 Semantic Characteristics of the ODP

Most categories in the ODP contain four parts, the *topic* of the category, *subcategories*, the *description* of the category, and a list of submitted Web pages, each with the *title* of the Web page, and a concise and accurate description of the submitted page. The information included in the category can be used to represent the semantic characteristics of the category, which can then be utilized to categorize and filter search results.

The *topic* of the category is actually the path from the root of the ODP hierarchy to the given category. It shows how to gradually narrow down from the most general concepts (the whole Web) to the more specific concepts represented by the category. Each category lower down the hierarchical structure represents a more specific concept. The concepts represented by supercategories are relevant to the given category.

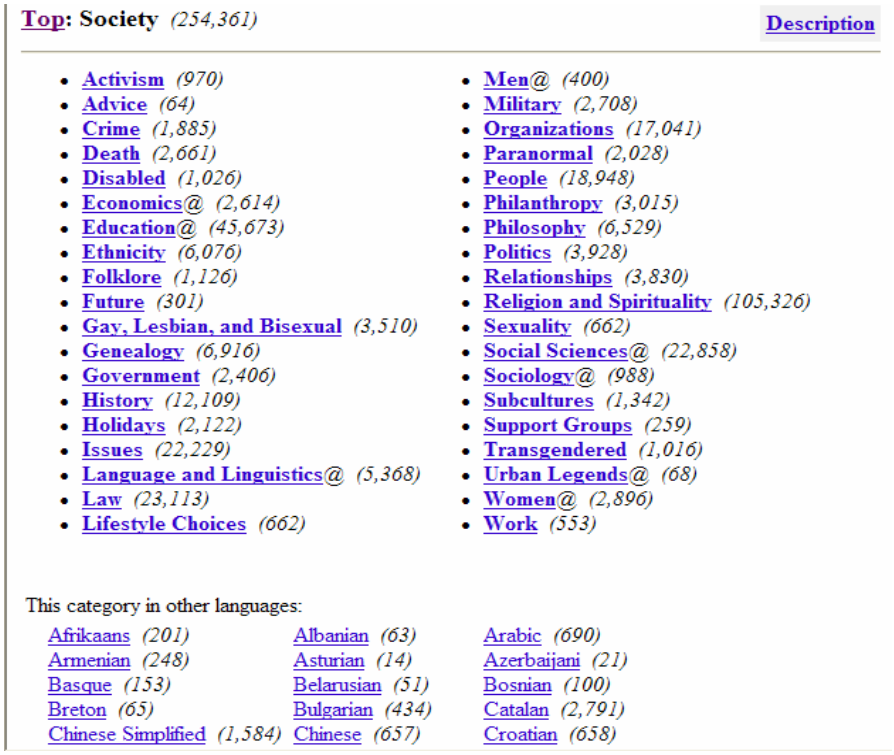


Figure 4: Subcategories of the Category "Society" in the ODP (www.dmoz.org)

Most categories have a "Description" link. The description of a category gives further explanation of the meaning of topic, what subcategories are included in this category, some information about the content and subject matter of the category, and sometimes editorial information to guide the Web submitters as to what kind of Web sites should not be submitted under this category. For example, the editorial information about "Recreation: Autos: Makes_and_Models" is:

"Please try and find the most specific subcategory that your site would be suited to.

Auto dealership sites should be submitted to the proper location in Regional. Dealership links will NOT be listed anywhere in Recreation/Autos.

If your site is selling products online, please submit your site to the proper subcategory of Shopping/Vehicles. Such sites will NOT be listed anywhere in Recreation/Autos."

As can be seen from the above citation, editorial information is not semantically related to the category because it only instructs how to find a proper category and not to include a category or to submit a Web page, and therefore is not used to represent the semantic characteristics of the category.

For each of the submitted Web pages under a category, in addition to the title to identify the site, there is also a concise and accurate informative description of the Web page which informs the end users what they will find when the site is visited. The title and the brief description are semantic characteristics of the Web page submitted under the category. The submitted Web pages under a category are actually a cluster of semantically related Web pages that are considered suitable to be classified under the category. They can therefore be used to represent the semantic characteristics of the category. Figure 5 shows a list of the submitted Web pages with their brief descriptions under the category "Science: Biology: Flora and Fauna".

Top Science Biology Flora and Fauna (20,469) Description

[A B C D E F G H I J K L M N C F Q R S T U V W X Y Z Complete List]
Animalia (6,602) Plantae (11,751)
Fungi (1,484) Protista (469)
Monera (126) Viruses (7)
See also
Recreation Outdoors Wildlife (747)
Science Environment Biodiversity (841)
This category in other languages
Chinese (3) Czech (60) Dutch (79)
German (337) Hebrew (18) Hungarian (5)
Japanese (359) Korean (1) Russian (4)
Scots Gaelic (7)
ARKive Images of Life on Earth - The Noah's Ark for the Internet era A global initiative gathering together films, photographs and audio recordings of endangered species from round the world
Biodiversity Explorer - Classification of all forms of life, with some photos Has a strong South African emphasis
BioImages - Virtual Fieldguide for UK Biodiversity - Large selection of pictures of organisms, mostly British in origin Images are an aid to identification, showing different stages, states and views of the organisms Search the database or follow a taxonomic tree
BioLib - Taxonomic tree of plants and animals with a photo gallery and some checklists for the Czech Republic and Slovakia In Czech and English
BiologyBase - Checklists, articles, and links for the world of life
Canada's Aquatic Environments - Information about the aquatic habitats, animals, and plants of this country

Figure 5: Category "Flora and Fauna" and the indexed Web pages (www.dmoz.org)

Two kinds of information in a category are not used to represent the semantic characteristics of the category. The first type is the name of the subcategories under the category. Each subcategory has its own semantic characteristics, its own description, and submitted Web pages. Therefore, using subcategories to represent the semantic characteristics introduces noise to both the category and its subcategories. Another type of information comes from the "FAQ" part of some categories. FAQ contains some useful information about where is the best place to submit a Web page.

Some information contained in the FAQ may semantically relate to the category. However, most of the information in the FAQ relates to other categories. This can be seen from Figure 6 where most of the information is irrelevant to the category.

Combining the topic of the category, the description of the category, and the submitted Web pages under this category (title and brief description of each page) can form a *category-document* that represents the semantic characteristics of the category. The following example demonstrates how to form a *category-document*.

Figure 7 is the screenshot of the description of the category "Flora and Fauna". The *category-document* of this category is composed of the following elements:

1) The topic of the category: "Science: Biology: Flora and Fauna";

2) The description of the category as demonstrated in Figure 7.;

3) The submitted Web pages with a brief description - Figure 5.

```
FAQ - Dmoz/Science/Biology/Flora_and_Fauna/Animalia
        Archieve-name          domz_org/Science/Biology/Flora_and_Fauna/Animalia
        Posting-Frequency      none
        Last-modified          2002-01-10 16 32 04
        URL                    http://dmoz org/Science/Biology/Flora_and_Fauna/Animalia/faq html
        Category               Science/Biology/Flora_and_Fauna/Animalia
Table of Contents
1 Where should I submit my web site about animals? Where will I find the animal topic I am looking for?

1 Q Where should I submit my web site about animals? Where will I find the animal topic I am looking for?
  A Please read
    http://dmoz org/Science/Environment/Biodiversity/faq htm
                                                                  by pst at 2002-02-12 a6 32 04

                                Help build the largest human-edited directory of the web
        Open Directory Home     http://dmoz org/
        About the Open Directory http://dmoz org/about html
        This FAQ                http://dmoz org/Science/Biology/Flora_and_Fauna/Animalia/faq html
        Open Directory Category Science/Biology/Flora_and_Fauna/Animalia
```

Figure 6: Information related to FAQ tag (www.dmoz.org)

3 The ODP Data

The ODP data is organized in two files, *structure.rdf* and *content.rdf*. The former contains category hierarchy information and the latter includes links within each category.

3.1 Data in *structure.rdf* File

The ODP is an open source project under the Open Directory Project Licence [Open Directory License 2007], and all the ODP data is downloadable from [The Open Directory Project 2008]. To illustrate the data structure of the ODP, *kt-structure.rdf.u8* (downloaded on 11 June 2006) is used as an example because it gives a comprehensive structure of the subcategory of "Kids and Teens". The subcategories of "Kids and Teens" are: Arts; Computers; Directories; Entertainment; Games; Health;

News; People and Society; Pre-School; School Time; Sports and Hobbies; Teen Life; Your Family; International. The "Description" of this category is shown in Figure 8.

Top Science Biology Flora and Fauna

This category is intended for websites about the biology of specific organisms or taxonomic groups The structure is organized according to a taxonomy tree, with the top-level subcategories being the five Kingdoms plus Viruses

Potential contents include descriptions and images of the organisms, classification, anatomy, physiology, behavior, distribution, reproduction and life cycle, habitat, biological or ecological aspects of management, endangered-species status, etc

Note In establishing the taxonomy-based category structure, some subtaxons and taxon levels are intentionally omitted for
1) escaping from 'unstable' taxons;
2) ease in navigation;
3) convenience in editing

Animalia
See this FAQ for advice on 'Where to submit or find a site about animals?'
The kingdom Animalia comprises all the creatures we normally think of as animals but it extends further than this, including worms, insects, crustaceans, molluscs, sponges, jellyfish as well as vertebrates Some animal-like organisms consisting of a single cell or a few cells are included in the category Protista The bacteria and the archaea are included in the category Monera

Figure 7: "Description" part of Category "Flora and Fauna" (www.dmoz.org)

Kids_and_Teens is targeted for audiences 18 and under Please note that this directory is for kids and not about kids For this reason we ask that you carefully follow our guidelines

Sites intended for a general audience may be accepted in Kids_and_Teens provided they are appropriate for individuals ages 18 and under Please note that no site is guaranteed placement in Kids_and_Teens
Sites containing chat rooms, discussion forums, or any other interactive content will be listed only if interactive content complies with Kids and Teens guidelines Please review your site carefully before submitting If any portion of your site contains profanity, obscenities, and/or sexually-explicit content, do not submit it to Kids and Teens Submission could result in its being banned altogether
Sites which are still under construction, submitted to multiple or inappropriate subcategories, using multiple URLs for multiple submissions, or consisting primarily of affiliate links will be deleted without review
If you believe your site complies with these guidelines, submit it once to the single most appropriate category

Kids and Teens is an Internet directory created especially for children and teenagers It includes both sites designed specifically for children and/or teens as well as sites designed for general audiences It does not include sites that are designed primarily to sell merchandise, sites that use profanity or obscenity, or sites that contain sexually explicit content

Figure 8: "Description" part of Category "Kids and Teens"(www.dmoz.org)

The *kt-structure.rdf* file has the following xml format (only a very small part of the file is presented here, and the lines are numbered for the purpose of explanation) as shown in Fig 9.

```
1)      <?xml version= 1 0  encoding= UTF-8' ?>
2)      <RDF xmlns r='http://www w3 org/TR/RDF/"
        xmlns d='http://purl org/dc/elements/1 0/"
        xmlns='http://dmoz org/rdf">
3)      < -- Generated at 2006-06-11 00 25 05 GMT on dust -->
4)      <Topic r id='Top/Kids_and_Teens">
5)      <catid>471237</catid>
6)      <d Title>Kids_and_Teens</d Title>
7)      <d Description>Kids and Teens is an Internet directory created especially for children
        and teenagers  It includes both sites designed specifically for children and/or …
        </d Description>
8)      <lastUpdate>2005-12-08 21 57 54</lastUpdate>
9)      <narrow1 r resource='Top/Kids_and_Teens/Pre-School"/>
10)     <narrow1 r resource='Top/Kids_and_Teens/Computers"/>
…
11)     </Topic>

12)     <Topic r id='Top/Kids_and_Teens/Pre-School">
13)     <catid>468769</catid>
14)     <d Title>Pre-School</d Title>
15)     <d Description>Sites in this category are aimed at children who cannot yet read  Subject
        matter … with the 6-7 year age group  </d Description>
…
16)     <narrow r resource='Top/Kids_and_Teens/Pre-School/Animals"/>
17)     <narrow r resource='Top/Kids_and_Teens/Pre-School/People"/>
…
18)     </Topic>
19)     <Topic r id='Top/Kids_and_Teens/Pre-School/Animals">
20)     <catid>1379018</catid>
21)     <d Title>Animals</d Title>
22)     <d Description>This category contains sites that … games  </d Description>
23)     <related r resource='Top/Kids_and_Teens/School_Time/Science/Living_Things/
        Animals"/>
24)     <related r resource='Top/Kids_and_Teens/Your_Family/Pets"/>
25)     <lastUpdate>2005-10-21 09 15 47</lastUpdate>
26)     <narrow2 r resource='Top/Kids_and_Teens/Pre-School/Animals/Dinosaurs"/>
27)     <narrow2 r resource='Top/Kids_and_Teens/Pre-School/Animals/Minibeasts"/>
28)     </Topic>
29)     <Topic r id='Top/Kids_and_Teens/Pre-School/Animals/Dinosaurs">
30)     <catid>1379335</catid>
31)     <d Title>Dinosaurs</d Title>
32)     <d Description>This category … stories, songs and games  </d Description>
33)     <related  resource='Top/Kids_and_Teens/School_Time/Science/The_Earth
        /Prehistoric_Times/Animals/Dinosaurs"/>
34)     <lastUpdate>2006-04-22 01 00 08</lastUpdate>
35)     </Topic>
…
```

Figure 9: Format of the ODP kt-structure.rdf (www.dmoz.org)

Lines 1 to 3 in Figure 9 are the head of the RDF file. Lines 4 to 11 are the XML description of the category "Kids and Teens" (note that category is called "Topic" in the RDF file). Elements of this category are enclosed between the <Topic> (line 4) and </Topic> tags (line 11). Each <Topic> tag has an r:id attribute (line 4) and encloses a <d:Description> tag (line 7), which encloses a topic description text. The <catid> (line 5) and <d:Title> (line 6) attributes stand for the category identifier and category title respectively. Text between <d:Description> and </d:Description> (line 7) is the description of the category. The content between lines 9 and 10 is the list of subcategories under the category "Kids and Teens", each of the subcategories is marked by a tag <narrow1 r:resource = "…" >.

Each of these subcategories under the category "Kids and Teens" has its own <Topic> and </Topic> pair which contains elements to describe these subcategories. Line 9 <narrow1 r:resource= "Top/Kids_and_Teens/Pre-School"/> indicates that category "Pre-School" is a subcategory of "Kids and Teens". Data between lines 12 to 18 is the description of this subcategory, with a similar structure to the category "Kids and Teens". Data between lines 19 to 28 is the description of the category "Animals" which is a subcategory of "Kids and Teens/Pre-School". Furthermore, lines 29 to 35 describe the category "Dinosaurs", which is the subcategory of "Animals".

Based on the description above it can be seen that each category (topic) name is built in the <Topic> tag's r:id attribute, that is, the path of the category follows directly after the tag "r:id=". Information between tags <d:Title> and </d:Title> only stands for the specific category (topic), and does not include its supercategories. The corresponding description information (if it has one) is enclosed within the tag pair <d:Description> and </d:Description>. Other elements, such as editorial information that contributes little to the semantics of the given category, are ignored in this research.

Because of the similarity of structure, the analysis process first identifies <Topic> and </Topic> pairs. Then, within each pair, extracting the topic name from the r:id tag, and the description text of this category, if there is a <d:Description> attribute. The category name and its corresponding description are organized into a Java key-value hash data structure (java.util.HashMap) for later use.

3.2 Data in *content.rdf* File

The *content.rdf* file shown in Figure 10 contains all topics, their links (submitted Web pages, or resources) and resource descriptions. For example, *kt-content.rdf* is used to illustrate how to extract the data from the *content.rdf* file. Only part of the *kt-content.rdf* file is depicted in Figure 10. The lines are numbered for the purpose of explanation. The *kt-content.rdf* file has the following xml format.

In Figure 10, lines 1 to 3 contain the RDF file head information. Data between lines 4 to 6 show that under the category "Kids and Teens", there are no external links. However, an empty template is provided. Information between lines 7 to 11 is an overview description of the links under category "Kids and Teens/Pre-School". Lines 7 and 8 give the name of the category and the category identifier respectively. From line 9 to 11 is the list of Web addresses of the external links under this category. Each of the elements has a format <link r:resource= "…" >. The title of each of these external links and their brief description are followed immediately after each <Topic>

and </Topic> pair. For example, content included between lines 12 to 17 provide the descriptive information of the Web site given in line 12, that is, the title of the Web site (line 13), the brief description of the Web site (line 14), and which category the Web site is submitted to (line 16). Information between line 18 to line 28 are similar to that of line 7 to line 17, which provide descriptive information of the external links under category "Kids and Teens/Pre-School/Animals", a subcategory of the category "Kids and Teens/Pre-School". The whole structure of the *content.rdf* file is similar to the structure described above.

```
1)    <?xml version= 1 0  encoding= UTF-8' ?>
2)    <RDF xmlns r="http://www w3 org/TR/RDF/"
      xmlns d="http://purl org/dc/elements/1 0/"
      xmlns="http://dmoz org/rdf">
3)    < -- Generated at 2006-06-11 00 25 05 GMT on dust -->
4)    <Topic r id="Top/Kids_and_Teens">
5)    <catid>471237</catid>
6)    </Topic>
7)    <Topic r id="Top/Kids_and_Teens/Pre-School">
8)    <catid>468769</catid>
9)    <link r resource="http://www enchantedlearning com/rhymes/painting/"/>
10)   <link r resource="http://www megafile com br/"/>
..
11)   </Topic>

12)   <ExternalPage about="http://www enchantedlearning com/rhymes/painting/">
13)   <d Title>Rebus Rhymes   EnchantedLearning com</d Title>
14)   <d Description>Preschoolers paint … they can read in their favorite rhymes </d Description>
15)   <ages>kids</ages>
16)   <topic>Top/Kids_and_Teens/Pre-School</topic>
17)   </ExternalPage>
..
18)   <Topic r id="Top/Kids_and_Teens/Pre-School/Animals">
19)   <catid>1379018</catid>
20)   <link r resource="http://www juliasrainbowcorner com/html/animalsmain html"/>
21)   <link r resource="http://www phonics jazzles com/html/freebie html"/>
..
22)   </Topic>
23)   <ExternalPage about="http://www juliasrainbowcorner com/html/animalsmain html">
24)   <d Title>Julia s Rainbow Corner  Animals</d Title>
25)   <d Description>Play games … and the noises they make </d Description>
26)   <ages>kids</ages>
27)   <topic>Top/Kids_and_Teens/Pre-School/Animals</topic>
28)   </ExternalPage>
..
```

Figure 10: Format of the ODP kt-content.rdf (www.dmoz.org)

According to the above, all the categories in the ODP with their descriptions (if the descriptions exist) are extracted from the *structure.rdf* file. The name of submitted Web pages and their brief descriptions are extracted from the *content.rdf* file. By

matching the topic in the two files, a text file, named as *category-document* for each category, is constructed for our purpose with the following three elements:

- the topic of the category;
- the description of the category;
- A list of submitted Web pages with their brief descriptions.

The full name of each topic (such as "Top/Kids_and_Teens/Pre-School/Animals") identifies different categories in the ODP. This is used to form the name for the textual *category-document*. Two minimal changes are made before using the topic name for naming the text file: first, note that the word "Top" is the same for all topics and contributes nothing for identifying the differences among categories and is therefore removed from the corresponding text file name. Second, the underscore character "_" is used as the separator between supercategory and subcategory instead of the slash ("/") character. To eliminate the confusion between the separator "_" with the underscore used in some of the ODP categories, such as "Kids_and_Teens", a pre-process is performed to change the underscore in the topic to a dash "-", that is, "Kids_and_Teens" should be changed to "Kids-and-Teens". After the two processes, the corresponding text file name for the topic "Top/Kids_and_Teens/Pre-School/Animals" is changed to "Kids-and-Teens_Pre-School_Animals". From the name of the category, it is clear that the first level category is "Kids_and_Teens", the second level category is "Pre-School", and the third level category is "Animals".

Furthermore, the constructed textual *category-document* can be easily organized into a tree-like file directory structure, just as the files organized in Microsoft Windows Explorer or Mac OS Finder. In fact, in this research, all the text files are stored in a directory tree structure which is the same as the directory tree structure of the ODP.

4 Implementation

There are two interfaces to parse an XML document: Object-Based, such as Document Object Model, and Event-Based, such as Simple API for XML (SAX) [Marchal 2000]. SAX models the parser, while DOM models the document. Both have their application situations as described below in Table 1 [Harold 2002].

In this research, the two XML documents, *content.rdf* and *structure.rdf* are very large. Therefore, SAX is selected to process the ODP data.

XML parser events include: element opening/closing tags, content of elements; entities; and parsing errors [Marchal 2000]. The event-handlers are registered by two designed classes corresponding to the two ODP data files: *StructureExtractor* and *ContentExtractor*.

StructureExtractor extracts topics and their descriptions. Each topic name is taken from the <Topic> tag's r:id attribute, and its corresponding description (if applicable) is taken from the text enclosed within the <d:Description> and </Description> tag pair. The extracted <Topic, Description> pair is stored in a Java *HashMap* structure.

ContentExtractor analyzes and extracts information from submitted Web pages under a given category. Each topic name is known by extracting the <Topic> tag's r:id attribute. All the <Link> tags within this <Topic> tag are parsed and their

r:resource attributes are read into a *list*. For each of the links stored in the *list*, there is a corresponding <ExternalPage> tag pair. For each of the tag pairs, the enclosed text within the <d:Title> and <d:Description> is taken and replaces the corresponding link in the *list*.

Table 1: Search terms used in the experiment

	SAX	DOM
When to use	• Document is large, and does not fit into available memory; • Document needs to be processed in small contiguous chunks of input; • Processing can be divided into a chain of successive operations.	• Multiple small documents need to be processed at the same time; • Internal data structure is simple; • The documents need to be modified repeatedly; • The documents need to be repeatedly stored in memory, and processed through many method calls.
Advantages	• Efficient, less memory requirement; fast.	• Simple, easy to grasp by programmers.
Disadvantages	• Needs more coding.	• DOM was designed by a committee trying to reconcile differences among the object models; • DOM is a lowest-common-denominator API that does not take full advantage of Java.

StructureExtractor provides category names (topic) and descriptions of the categories; *ContentExtractor* obtains category names, and a list of submitted Web pages with their brief descriptions. All of this information is used to build a textual *category-document* set described in the Section 3, The ODP Data.

The UML of the two classes *structureExtractor* and *contentExtractor* are depicted in Figure 11. As can be seen from this figure, our two essential classes, *structureExtractor* and *contentExtractor* are extended from the class *org::mxl::sax::helpers::DefaultHandler,* which provides default implementations for the core functions of SAX. Class *ODPExtractor* instantiates the above two classes to realise the semantic characteristics extracted. *TopicClass* extracts the topics from the ODP. The Java code of *structureExtractor* is shown in Figure 12. Lines 3 and 4 define two String variables “topic” and “currentDescription” which are used to store a given category and the semantic characteristics of the category. Line 7 indicates the extracted contents are stored in a HashMap structure. startElement() method between line 16 and 19 extracts topic; endElement() method between line 20 and 27 put the

<topic, currentDescription > pair into the HashMap structure; and characters() method concatenates the relevant contents into "currentDescription".

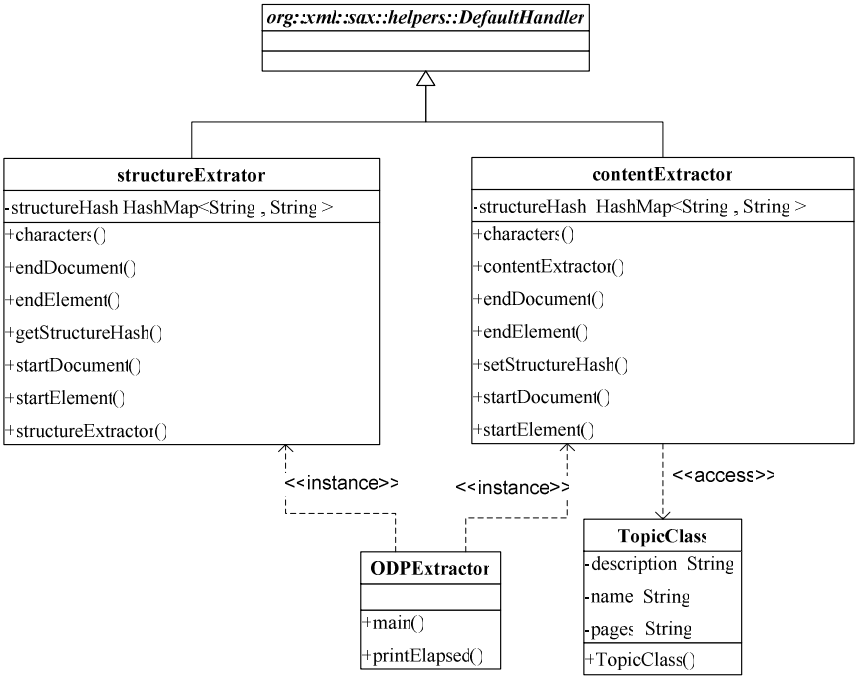


Figure 11: UML of class structureExtractor and contentExtractor

5 Application and Experimental Results

5.1 Application

The *category-document* set can be used to, for example, classify Web search results [Zhu 2007, Zhu and Dreher 2007]. We use Lucene [Gospodnetić and Hatcher 2005] to implement our system. Stop words are first removed from the *category-document set* and *Websnippet* set; they are then stemmed to further reduce the dimensionality of the vector space, within which the cosine similarities between the vectors representing *Websnippet* set and the vectors of *category-document set* are compared.

The content of *category-document* and *Websnippet* can be represented by a set of indexed terms t_i (i = 1,2,…t), or term vector $\mathbf{d_j} = (t_1,t_2,…t_t)$. Not all terms are equally important to describe the content of a document. Term-frequency - reverse document frequency (tf-idf) is the most popular way to assign a weight to the terms in a document [Baeza-Yates and Ribeiro-Neto 1999, Salton and Buckley 1988]. Therefore, a *category-document* $\mathbf{d_j}$ can be represented by a t-dimensional vector $\mathbf{d_j} = (w_{1,j}, w_{2,j}, \ldots w_{t,j})$, a *Websnippet* can also be represented as a t-dimensional vector $\mathbf{q} = (w_{1,q},$

$w_{2,q}$, … $w_{t,q}$). $w_{i,j}$ (term i in document j) is calculated by the following formulas, where max*tf* is the maximum number of term appears in document $\mathbf{d_j}$, or query q:

$$w_{i,j} = (0.5 + \frac{0.5tf}{\max tf}) \bullet \log \frac{N}{n_i}$$

$$w_{i,q} = 0.5 + \frac{0.5tf}{\max tf}$$

```
1)      public class structureExtract extends DefaultHandler
2)      {
3)              String currentTopic = "";
4)              String currentDescription = "";
5)              Stack<String> xmlStack = new Stack<String>();
6)              int numTopics = 0;
7)              HashMap<String,String> structureHash = new HashMap<String,String>(1000000,
                                                                                    0 75f);

8)              public structureExtract() {
9)                      super(); }

10)             public HashMap<String,String> getStructureHash() {
11)                     return structureHash;       }

12)             public void startDocument(){
13)                     System out println("Parsing structure rdf...");    }

14)             public void endDocument() {
15)                     System out println(numTopics+' topics parsed.");   }

16)             public void startElement(String namespaceURI, String localName, String qName,
                                Attributes atts)
17)             {       xmlStack push(qName);
18)                     if( qName equals(' Topic")){
19)                             currentTopic = atts getValue("r id"); //returns null if there is no r id } }

20)             public void endElement(String uri, String localName, String qName){
21)                     if( qName equals(' Topic")) {
22)                             if(!( currentTopic equals("") || currentTopic equals(null) ||       1)
                                                                    currentTopic equals("Top:"))) {
23)                                     structureHash put(currentTopic,currentDescription);
24)                                     currentDescription = "";
25)                                     numTopics++; }
26)                             currentTopic = ""; }
27)                     xmlStack pop(); }

28)             public void characters (char ch[], int start, int length) {
29)                     for (int i = start; i < start + length; i++) {
30)                             switch (ch[i]) {
31)                                     case '\n   ch[i] = ' '; break;
32)                                     case '\r   ch[i] = ' '; break;
33)                                     case '\t   ch[i] = ' '; break; } }
34)                     if( xmlStack peek() equals(' d Description") ) {
35)                             currentDescription = currentDescription concat( new String(ch, start,
                                                                                    length)); }
36)             }
37)     }
```

Figure 12: Java code for implementing StructureExtractor

The degree of similarity between *category-document set and Websnippet* set can be calculated by the cosine value of the angle (θ) between these two vectors. According to the definition of dot product of two vectors in a vector space **A** ● **B** =

$|\mathbf{A}| \times |\mathbf{B}| \cos(\theta)$, we have the following formula is [Baeza-Yates and Ribeiro-Neto 1999]:

$$\mathrm{Sim}(\mathrm{d_j}, \mathrm{q}) = \cos(\theta) = \frac{\mathrm{d_j} \bullet \mathrm{q}}{|\mathrm{d_j}| \times |\mathrm{q}|} = \frac{\sum_{i=1\ldots N} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1\ldots N} w_{i,j}^2 \times w_{i,q}^2}}$$

The *Websnippet* objects are to be organized under the first two levels of the ODP, with 588 categories. Because there are a total of 59,000 categories, each of the categories of the first two levels will contain hundreds or thousands of *category-document objects*; that is, the semantic content of the category is manifested by the *category-document set* belonging to it.

For a given *Websnippet*, the ODP categories represented by *category-document sets* are ranked according to the descending order of the calculated cosine similarities between the vector representing the *Websnippet* and the ODP category vectors representing the objects in the *category-document sets*. Considering the first *k* ODP categories with the highest cosine scores, the given *Websnippet* is to be assigned to the most frequently occurring category computed according to the majority voting algorithm as shown in Figure 13 [Zhu 2007]. This is our implementation of kNN.

5.2 Experimental Results

Precision and *recall* are the two most widely accepted and used measurements of the retrieval performance of an information retrieval system. *Recall* is a criterion to measure the ability of an information retrieval system to retrieve **all** relevant documents; *precision* measures the ability of an information retrieval system to retrieve **only** relevant material, as defined below:

> *… information query I (of a test reference collection) and its set R of relevant documents. Let |R| be the number of documents in this set. Assume that a given retrieval strategy (which is being evaluated) processes the information request I and generates a document answer set A. Let |A| be the number of documents in this set. Further, let |Ra| be the number of documents in the intersection of the sets R and A (that is, |Ra| is the number of relevant documents in the intersection of the sets R and A).*
>
> *Recall is the fraction of the relevant documents (the set R) which has been retrieved,*
>
> $$Recall = |Ra| / |R|$$
>
> *Precision is the fraction of the retrieved documents (the set A) which is relevant,*
>
> $$Precision = |Ra| / |A|$$

[Baeza-Yates and Ribeiro-Neto 1999]

For text categorization, a similar definition is given by [Yang 1999]:

$$Precision = \frac{\text{categories found and correct}}{\text{total categories found}}$$

$$Recall = \frac{\text{categories found and correct}}{\text{total categories correct}}$$

Search results are presented in a ranked list according to the degree of relevance of the document to a given query. Users then examine the ranked list starting from the top of this list. Thus, the *recall* and *precision* measures vary as the users proceed with their examination of the retrieved answer set. To evaluate the ranked lists, *precision* is plotted against *recall* after each retrieved document – a standard 11 points *precision* versus *recall* is usually used as an overall evaluation of search results [Baeza-Yates and Ribeiro-Neto 1999].

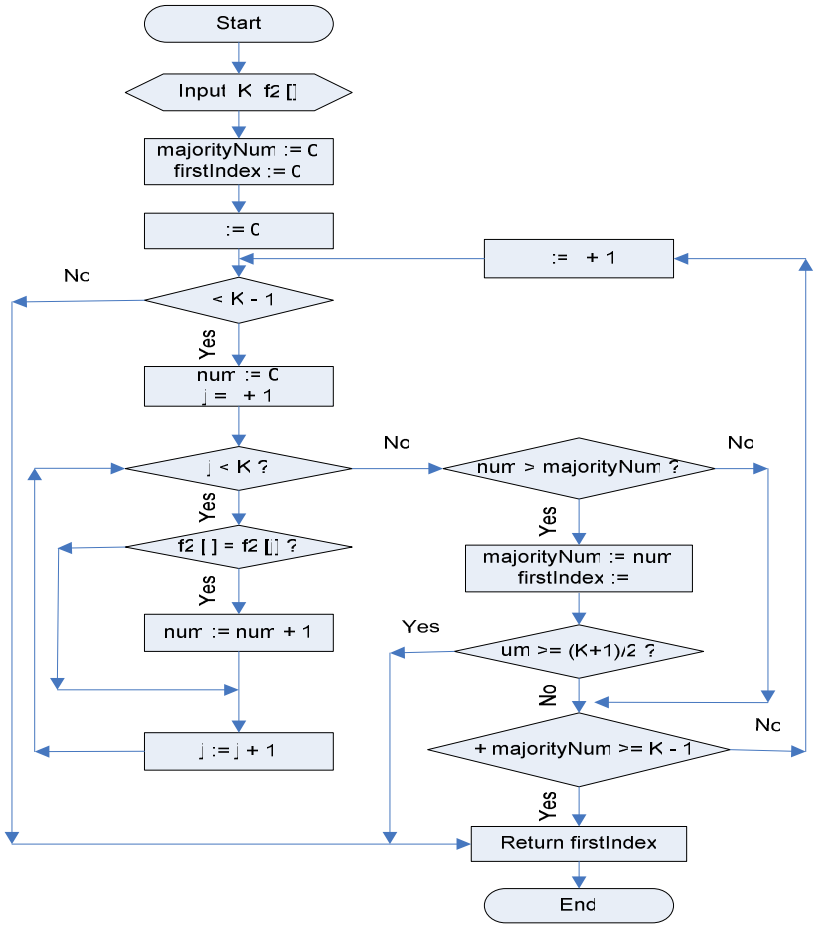


Figure 13: Majority Voting processing flowchart [Zhu 2007]

In Web search, it is impractical to find all relevant and irrelevant documents for a given query. It is thus impossible to calculate *precision* and *recall*. TREC, Text REtrieval Conference (http://trec.nist.gov), uses *recall* and *precision* at various cut-off levels to compare the performance of an IR system [Hawking, et al. 1999]. A cut-off level is a rank that defines the retrieved set. For example, a cut-off level of 10 defines the top ten retrieved documents in the ranked list. If seven out of the ten returned documents are relevant, the *precision* at cut-off level ten (P@10) is then 7/10 = 0.7 = 70 per cent.

To evaluate the performance of our proposed approach, we consider only the retrieved document set, the calculation of *precision* and *recall* are based on this set. We choose only the traditional information retrieval measurements: the standard 11 points *precision-recall* curve, and TREC style *precision* measurement.

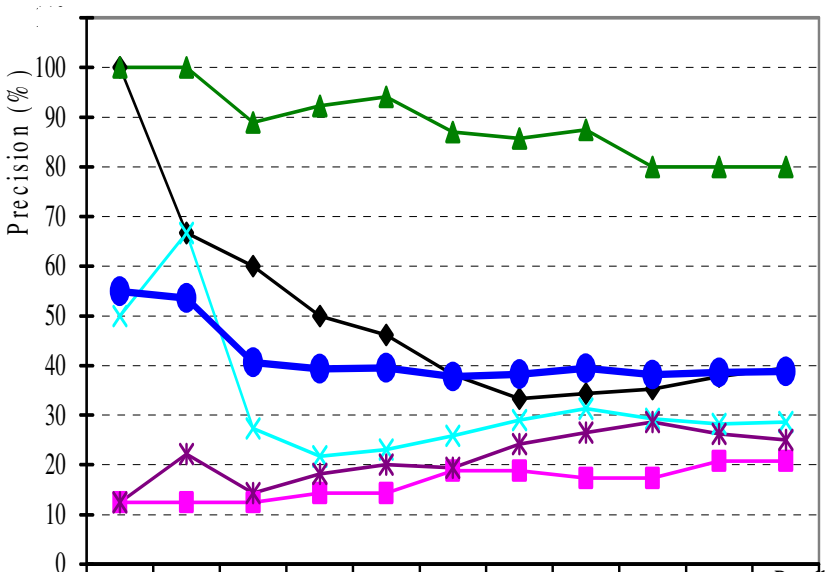
Our experiment is designed as follows. First, a Special Search Browser (SSB) has been developed which implements kNN text categorization algorithms by using Lucene. Web search results are obtained from Yahoo Search Web Services APIs. Yahoo APIs rather than Google Web APIs are utilized because when SSB was developed, Yahoo APIs return 50 search results for each query, and Google APIs return only ten search results for each query. Ten results are obviously not sufficient for categorization research purposes. Five ambiguous or general queries are selected and submitted to Yahoo, and 50 *Websnippet* objects are returned for each of the queries. For each query, a clear information need is defined. Five human experts are employed to decide if the returned *Websnippet* set is relevant to the given information needs. Human experts know nothing about our proposed approach and are only presented with a list of returned *Websnippet* sets and the information needs. They are asked to judge if the *Websnippet* objects are relevant to the specified information needs.

Meanwhile, the objects in the *Websnippet* set are also categorized according to ODP categories and we decided that the two categories with the greatest number of relevant search results would be sufficient for SSB to satisfy users' information needs. These search results are therefore selected as the categorized results of SSB.

Experimental results demonstrate using extracted semantic characteristics of the ODP categories to classify *Websnippet* can improve the *precision* of the Web searching by more than 23 percent. The improvement of P@5 and P@10 is more than 30% on average [Zhu 2007]. Table 2 is the search-terms used in the experiment. Three types of queries are utilized in our experiment. The first type is ambiguous queries which have more than one meaning; the second type of queries are entity names, each entity name usually indicates a person's name but it could be a place or other object; the third type of queries are general terms that have general meanings, such as health. Because all of the queries have more than one meaning, a definite information need is specified for each of the queries as shown in the "Information need" column in Table 2. Figure 14 is the *precision-recall* curve [Voorhees 2005] of the search results of Yahoo API, Figure 15 is the matching *precision-recall* curve of the categorized results of SSB. Figure 16 compares the averaged search results of Yahoo and the proposed approach (SSB).

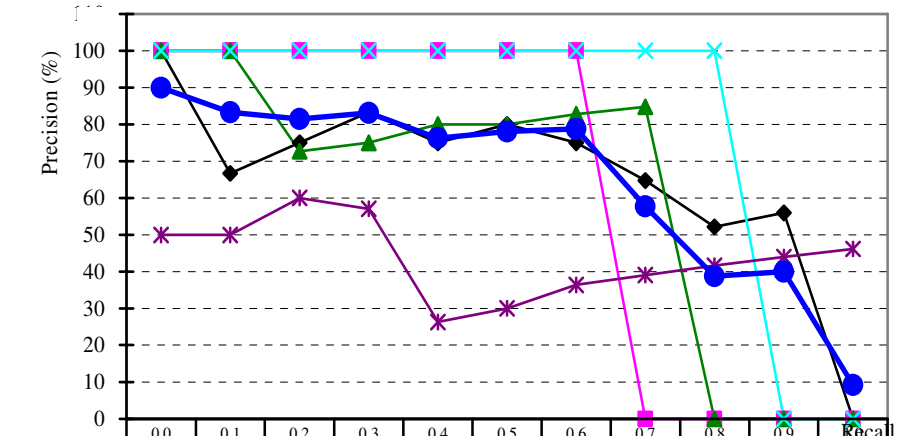
Table 2: Search terms used in the experiment

Query type [Zeng, et al. 2004]	**Query**	**Information need**
Ambiguous query	jaguar	Information about the animal jaguar
	UPS	Information about how UPS (Uninterruptible Power Supply) works; key specification of UPS
Entity name	Clinton	The American president William J. Clinton
	Ford	Henry Ford, the founder of the Ford Motor Company
General term	health	How can one keep healthy



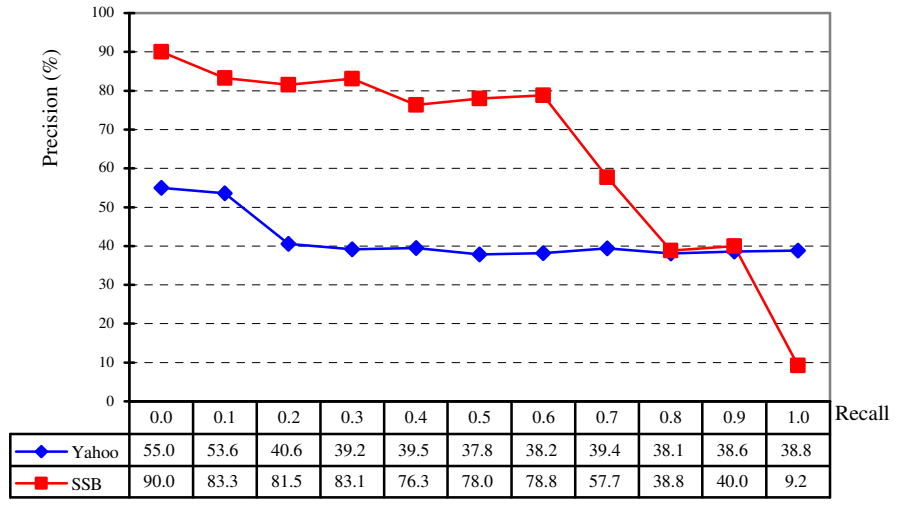
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Clinton	100.0	66.7	60.0	50.0	46.2	38.1	33.3	34.4	35.3	37.8	39.5
Ford	12.5	12.5	12.5	14.3	14.3	18.8	18.8	17.4	17.4	20.8	20.8
Health	100.0	100.0	88.9	92.3	94.1	87.0	85.7	87.5	80.0	80.0	80.0
Jaguar	50.0	66.7	27.3	21.7	23.1	25.9	29.0	31.3	29.3	28.3	28.6
UPS	12.5	22.2	14.3	18.2	20.0	19.4	24.2	26.5	28.6	26.2	25.0
Average	55.0	53.6	40.6	39.3	39.5	37.8	38.2	39.4	38.1	38.6	38.8

*Figure 14: Precision-recall curve for search results of **Yahoo** for each of the five search-terms*



	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Recall
Clinton	100.0	66.7	75.0	83.3	75.0	80.0	75.0	64.7	52.2	56.0	0.0
Ford	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0
Health	100.0	100.0	72.7	75.0	80.0	80.0	82.8	84.8	0.0	0.0	0.0
Jaguar	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	0.0
UPS	50.0	50.0	60.0	57.1	26.3	30.0	36.4	39.1	41.7	44.0	46.2
Average	90.0	83.3	81.5	83.1	76.3	78.0	78.8	57.7	38.8	40.0	9.2

*Figure 15: Precision-recall curve for search results of **SSB** for each of the five search-terms*



	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	Recall
Yahoo	55.0	53.6	40.6	39.2	39.5	37.8	38.2	39.4	38.1	38.6	38.8	
SSB	90.0	83.3	81.5	83.1	76.3	78.0	78.8	57.7	38.8	40.0	9.2	

*Figure 16: Average recall-precision curves of **Yahoo!** search results and **SSB Categorized** search results over the five search-terms*

The comparison of P@5, P@10 of search results of Yahoo and the proposed method (SSB Categorized) is shown in Table 3.

*Table 3: P@5 and P@10 of **Yahoo!** and **SSB Categorized** search results (%)*

	P@5	**P@10**	**Average**
Yahoo!	46.7	42.0	44.4
SSB Categorized	85.0	70.0	77.5
Improvement	38.3	28.0	33.2

6 Limitations and Discussion

6.1 Limitations of Using the ODP

Some Web pages are not submitted to the appropriate categories in the ODP. This may happen when the submitter of the Web pages notices only the meaning expressed by the title of the subcategory, but fails to notice the meaning expressed by the "topic" of the category.

For example, under the category "Society/History/Education", a Web page titled "Y-Vote Mock Elections" was submitted. However, the page aims to activate students by giving them the opportunity to stand as party candidates, or as speechwriters in a mock election. Obviously, this page is education-related, but not really relevant to the category "History". This kind of problem is beyond the scope of this research and is thus not addressed.

According to [Labrou and Finin 1999, The Open Directory Project 2008], there are allegations that volunteer ODP editors treat their own Web pages on high priority thereby thwarting the efforts of their competition. However, this research mainly concerns whether the submitted Web pages under a category are relevant to the category, and gives less attention to anything that is irrelevant to process of semantic characteristics extraction. Therefore, this criticism has little impact on the semantic characteristics extraction process of each category.

6.2 Categorizing *Websnippet* into Only One Category

At this research stage, one *Websnippet* is categorized into only one category. This results in the following limitations.

1. The Web category is *per se* ambiguous. For example, the ODP category is different from the *Yahoo! Web Directory*. Notice that even experts have no agreement on how to classify search results; it is therefore impossible for an algorithm to categorize all the relevant documents into only one category. Further, relevance judgment *per se* is subjective, and varies at different times [Mizzaro 1997].
2. Search results are given in the form of a *Websnippet* set which does not purport to represent the semantic characteristics of the search results. Therefore, the information may not be fine grained enough to allow the proposed approach to classify the research results into proper categories.

3. The extracted semantic characteristics of the ODP category may be inaccurate, or insufficient to describe the concept represented by the category. In this circumstance, categorization may also be affected.

6.3 Other Issues

kNN is not as efficient as naïve Bayes and SVMs [Chakrabarti 2003, Sebastiani 2002] although it is a lazy learning algorithm which has no separate learning process, and it is only a little less effective than the top performing one, such as SVMs. It would be better to compare the effectiveness and efficiency of the different text categorization algorithms in case only a *Websnippet* set is available. It is also useful to explore the effectiveness of different feature selection/extraction algorithms [Yang and Pedersen 1997] in this research scenario.

In our experiment, only five search results are provided, and each query has 50 search results. This is a small sample, however, in this case each human expert has to make 5*50 = 250 relevance judgments for the items in the returned *Websnippet* set. In addition, 100 results for each query should be more statistically sound for our experimental results. Note that human experts only make relevance judgements, they are not asked to classify each *Websnippet* into the ODP categories. This will greatly reduce the human workload.

7 Conclusion

This paper discussed how to use Simple API for XML to extract semantic characteristics of the ODP categories to create *category-document* sets. The topic of each category, the description of the category, and the submitted Web pages with their brief descriptions under the category, comprise a *category-document*, which is consequently used to represent the semantic aspects of the category. Applying the *category-document* set as training data to categorize *Websnippet*, our experimental results reveal that the proposed approach can significantly improve *precision* of Web search results by more than 23 percent. This indicates that the ODP metadata, when aggregated, can represent the semantics of the ODP topics (categories), for use in boosting information retrieval from the Web.

It is also worth noting that the proposed approach automatically generates a *category-document* set by analyzing and extracting the semantic characteristics of ODP metadata. This automatically created *category-document* set is then utilized as training data. This approach can save us manually generating a training data set by employing human experts to label a *Websnippet* set into appropriate topics (categories).

Acknowledgements

This work is partially supported by Digital Ecosystems and Business Intelligence (DEBI) Institute of Curtin University. Thanks to Professor Elizabeth Chang for her continuous support and encouragement. Thanks also to Mr. Chris Jones for assistance with the programming.

References

[Baeza-Yates and Ribeiro-Neto 1999] Baeza-Yates R., and Ribeiro-Neto B. Modern Information Retrieval. Harlow: Addison Wesley, 1999.

[Chakrabarti 2003] Chakrabarti S. Mining the Web: Discovering Knowledge from Hypertext Data. Edited by Jim Gray, Data Management Systems. San Francisco: Morgan Kaufmann, 2003.

[Chirita, et al. 2005] Chirita P.-A., Nejdl W., Paiu R., and Kohlschütter C. "Using Odp Metadata to Personalize Search." In the Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005, 178-185.

[Deerwester, et al. 1990] Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., and Harshman R. "Indexing by Latent Semantic Indexing." Journal of the American Society for Information Science 41, no. 6 (1990): 391-407.

[Deng, et al. 2004] Deng Z.-H., Tang S.-W., Yang D.-Q., Zhang M., Li L.-Y., and Xie K.-Q. "A Comparative Study on Feature Weight in Text Categorization." In the Proceedings of Advanced Web Technologies and Application, 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, April 14-17, 2004, 588-597.

[Gauch, et al. 2003] Gauch S., Chaffee J., and Pretschner A. "Ontology-Based Personalized Search and Browsing." Web intelligence and Agent System 1, no. 3-4 (2003): 219-234.

[Gospodnetić and Hatcher 2005] Gospodnetić O., and Hatcher E. Lucene in Action. Greenwich: Manning Publications, 2005.

[Harold 2002] Harold E. R. Processing Xml with Java: A Guide to Sax, Dom, Jdom, Jaxp, and Trax. Boston, MA: Addison Wesley, 2002.

[Hawking, et al. 1999] Hawking D., Craswell N., Thistlewaite P., and Harman D. K. "Results and Challenges in Web Search Evaluation." Computer Networks 31, no. 11-16 (1999): 1321-1330.

[Jain, et al. 1999] Jain A. K., Murty M. N., and Flynn P. J. "Data Clustering: A Review." ACM Computing Surveys 31, no. 3 (1999): 264-323.

[Labrou and Finin 1999] Labrou Y., and Finin T. "Yahoo! As an Ontology - Using Yahoo! Categories to Describe Documents." In the Proceedings of the eighth international conference on Information and knowledge management, Kansas City, MO, USA., 1999, 180-187.

[Lewis, et al. 2004] Lewis D. D., Yang Y., Rose T. G., and Li F. "Rcv1: A New Benchmark Collection for Text Categorization Research." Journal of Machine Learning Research 5 (2004): 361-397.

[Marchal 2000] Marchal B. Xml by Example. Indianapolis: QUE, 2000.

[Mitchell 1997] Mitchell T. M. Machine Learning. New York: McGraw-Hill Companies, 1997.

[Mizzaro 1997] Mizzaro S. "Relevance: The Whole History." Journal of the American Society for Information Science 48, no. 9 (1997): 810-832.

[Mladenic 1998] Mladenic D. "Turning Yahoo into an Automatic Web-Page Classifier." In the Proceedings of the 13th European Conference on Artificial Intelligence, Yong Research Paper, Brighton, UK, August 23-28, 1998, 473-474.

[Open Directory License 2007] Open Directory License. 2007.

http://www.dmoz.org/license.html. (accessed November 27, 2007, 2007).

[The Open Directory Project 2008] The Open Directory Project. 2008. www.dmoz.org. (accessed November 14, 2008, 2008).

[Perugini 2008] Perugini S. "Symbolic Links in the Open Directory Project." Information Processing and Management 44, no. 2 (2008): 910-930.

[Salton and Buckley 1988] Salton G., and Buckley C. "Term-Weighting Approaches in Automatic Text Retrieval." Information Processing & Management 24, no. 5 (1988): 513-523.

[Sebastiani 2002] Sebastiani F. "Machine Learning in Automated Text Categorization." ACM Computing Surveys 34, no. 1 (2002): 1-47.

[Sherman 2000] Sherman C. "Humans Do It Better: Inside the Open Directory Project." Online, July, 2000 2000.

[Turtle and Croft 1996] Turtle H. R., and Croft W. B. "Uncertainty in Information Retrieval Systems." In Uncertainty Management in Information Systems, edited by Amihai. Motro and Philippe Smets, 189-224. Boston: Kluwer Academic Publishers, 1996.

[Voorhees 2005] Voorhees E. M. "Common Evaluation Measures." In the Proceedings of the Fourteenth Text Retrieval Conference (TREC 2005), Gaithersburg, Maryland, November 15-18, 2005,

[Yang 1999] Yang Y. "An Evaluation of Statistical Approaches to Text Categorization." Information Retrieval 1, no. 2 (1999): 69-90.

[Yang and Liu 1999] Yang Y., and Liu X. "A Re-Examination of Text Categorization Methods." In the Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR '99), Berkeley, CA, August 15-19, 1999, 42-49.

[Yang and Pedersen 1997] Yang Y., and Pedersen J. O. "A Comparative Study on Feature Selection in Text Categorization." In the Proceedings of the 14th International Conference on Machine Learning (ICML-97), San Francisco, July 8-12, 1997, 412-420.

[Zeng, et al. 2004] Zeng H.-J., He Q.-C., Chen Z., Ma W.-Y., and Ma J. "Learning to Cluster Web Search Results." In the Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, 210-217.

[Zhu 2007] Zhu D. "Improving the Relevance of Search Results Via Search-Term Disambiguation and Ontological Filtering." Master dissertation, Curtin University of Technology, 2007.

[Zhu and Dreher 2007] Zhu D., and Dreher H. "Determining and Satisfying Search Users Real Needs Via Socially Constructed Search Concept Classification." In the Proceedings of the Inaugural IEEE Digital Ecosystems and Technologies Conference, Cairns, Australia, February 21-23, 2007, 404-409.

[Zhu and Dreher 2008] Zhu, D., and Dreher, H. "Improving Web Search by Categorization, Clustering, and Personalization." In LNAI 5139, Advanced Data Mining and Applications, the Fourth International Conference ADMA, edited by Tang, C. 659-666. Berlin, Heidelberg: Springer Verlag, 2008.