LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

Department of Industrial Engineering and Management

Thesis for the Master of Science in Technology

Text Classification and Indexing in IP Networks

The subject for this thesis has been approved by the council of the Department of Industrial Engineering and Management on August 23, 2000.

Supervisor: professor Markku Tuominen

Instructor: M.Sc. Lasse Metso

Lappeenranta, 20. October 2000

Kirsi Lehtinen

Kylväjänkatu 9

53500 LAPPEENRANTA

+358 5 416 1170

ABSTRACT

Author: Kirsi Lehtinen

Name of the Thesis: Text Classification and Indexing in IP Networks

Department: Industrial Engineering and Management

Year: 2000 **Place:** Lappeenranta

Master's Thesis. Lappeenranta University of Technology.

105 pages, 15 pictures, 3 tables and 3 appendices.

Supervisor professor Markku Tuominen.

Keywords: classification, indexing, information retrieval, Internet, IP, search engines, service management

Avainsanat: hakukoneet, indeksointi, Internet, IP, luokittelu, palvelun hallinta, tiedon haku

Internet is an infrastructure for electronic mail and has been an important tool for academic users. It has increasingly become a vital information resource for commercial enterprises to keep in touch with their customers and competitors. The increase in volume and diversity of WWW creates an increasing demand from its users of sophisticated information and knowledge management services. Such services include things like cataloguing and classification, resource discovery and filtering, personalization of access and monitoring of new and changing resources. Though the number of professional and commercially valuable information resources available on the WWW has grown considerably over the last years, they still rely on general-purpose Internet search engines. Satisfying the varied requirements of users for searching and retrieving documents have become a complex task to Internet search engines. Classification and indexing is an important part of the problem of accurate searching and retrieving. This thesis will introduce the basic methods of classification and indexing and some of the latest applications and projects in where the idea has been to solve that problem.

TIIVISTELMÄ

Tekijä: Kirsi Lehtinen

Tutkielman nimi: Tekstin luokittelu ja indeksointi IP-verkoissa

Osasto: Tuotantotalouden osasto

Vuosi: 2000 **Paikka**: Lappeenranta

Diplomityö. Lappeenrannan teknillinen korkeakoulu.
105 sivua, 15 kuvaa, 3 taulukkoa ja 3 liitettä.
Tarkastajana professori Markku Tuominen.

Avainsanat: hakukoneet, indeksointi, Internet, IP, luokittelu, palvelun hallinta, tiedon haku
Keywords: classification, indexing, information retrieval, Internet, IP, search engines, service management

Internet on elektronisen postin perusrakenne ja ollut tärkeä tiedonlähde akateemisille käyttäjille jo pitkään. Siitä on tullut merkittävä tietolähde kaupallisille yrityksille niiden pyrkiessä pitämään yhteyttä asiakkaisiinsa ja seuraamaan kilpailijoitansa. WWW:n kasvu sekä määrällisesti että sen moninaisuus on luonut kasvavan kysynnän kehittyneille tiedonhallintapalveluille. Tällaisia palveluja ovet ryhmittely ja luokittelu, tiedon löytäminen ja suodattaminen sekä lähteiden käytön personointi ja seuranta. Vaikka WWW:stä saatavan tieteellisen ja kaupallisesti arvokkaan tiedon määrä on huomattavasti kasvanut viime vuosina sen etsiminen ja löytyminen on edelleen tavanomaisen Internet hakukoneen varassa. Tietojen hakuun kohdistuvien kasvavien ja muuttuvien tarpeiden tyydyttämisestä on tullut monimutkainen tehtävä Internet hakukoneille. Luokittelu ja indeksointi ovat merkittävä osa luotettavan ja täsmällisen tiedon etsimisessä ja löytämisessä. Tämä diplomityö esittelee luokittelussa ja indeksoinnissa käytettävät yleisimmät menetelmät ja niitä käyttäviä sovelluksia ja projekteja, joissa tiedon hakuun liittyvät ongelmat on pyritty ratkaisemaan.

ACKNOWLEDGEMENTS

This thesis is made in Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology in Lappeenranta as a part of IPMAN-project.

I'd like to thank my instructor Lasse Metso for his valuable advice during the hole work and Ossi Taipale and Liisa Uosukainen from Taipale Engineering Ltd., those who made this work possible and helped me in the start. Also, I'd like to express my gratitude to my supervisor professor Markku Tuominen for his time and advises. A thanks belongs also to Ms. Barbara Cash for her time and advice in English language, and to IPMAN project manager Stiina Ylänen for her comments and assessments.

In the end I'd like to present the greatest thanks to my husband Esa and our two sons, Tuomas and Aapo for the patience, encouragement and support during my student years and this work.

Table of Contents

LIST OF FIGURES AND TABLES

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AI	Artificial Intelligence
CERN	European Organization for Nuclear Research
CGI	Common Gateway Interface
DARPA	Defense Advanced Research Projects Agency
DDC	Dewey Decimal Classification
DESIRE	Development of a European Service for Information on Research and Education
DFG	Deutsche Forchungsgemeinschaft
DTD	Document Type Definition
Ei	Engineering information
eLib	Electronic Library
ETH	Eidgenössische Technische Hochschule
GERHARD	German Harvest Automated Retrieval and Directory
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IR	Information Retrieval
ISBN	International Standard Book Numbers
KB	Knowledge base
LCC	Library of Congress Classification
LC	Library of Congress
MARC	Machine-Readable Cataloguing
MLP	Multi-Layer Perceptron Network
NCSA	National Center for Supercomputing Applications
PCDATA	parsed character data
RDF	Resource Description Framework
SIC	Standard Industrial Classification
SOM	Self-Organizing-Map
SGML	Standard General Markup Language
TCP	Transmission Control Protocol

UDC	Universal Decimal Classification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WEBSOM	Neural network (SOM) software product
VRML	Virtual Reality Modeling Language
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible MarkUp Language
XSL	Extensible Stylesheet Language
Xlink	Extensible Linking Language
Xpointer	Extensible Pointer Language

1 INTRODUCTION

The Internet, and especially its most famous offspring, the World Wide Web (WWW), has changed the way most of us do business and go about our daily working lives. In the past several years, the increase of personal computers and other key technologies such as client-server computing, standardized communications protocols (TCP/IP, HTTP), Web browsers, and corporate intranets have dramatically changed the manner we discover, view, obtain, and exploit information. As well as an infrastructure for electronic mail and a playground for academic users, the Internet has increasingly become a vital information resource for commercial enterprises, which want to keep in touch with their existing customers or reach new customers with new online product offerings. The Internet has also become an information resource for enterprises to keep clear about their competitor's strengths and weaknesses. (Ferguson and Wooldridge, 1997)

The increase in volume and diversity of the WWW creates an increasing demand from its users of sophisticated information and knowledge management services, beyond searching and retrieving. Such services include cataloguing and classification, resource discovery and filtering, personalization of access and monitoring of new and changing resources, among others. The number of professional and commercially valuable information resources available on the WWW has grown considerably over the last years, still relying on general-purpose Internet search engines. Satisfying the vast and varied requirements of corporate users is quickly becoming a complex task to Internet search engines. (Ferguson and Wooldridge, 1997)

Every day the WWW grows by roughly a million electronic pages, adding to the hundreds of millions already on-line. This volume of information is loosely held together by more than a billion connections, called hyperlinks.
(Chakrabarti et al. 1999)

Because of the Web's rapid, chaotic growth, it lacks organization and structure. People from any background, education, culture, interest and motivation with of many kinds of dialect or style can write Web pages in any language. Each page might range from a few characters to a few hundred thousand, containing truth, falsehood, wisdom, propaganda or sheer nonsense. The discovery of high-quality, relevant pages in response to a specific need for certain information from this digital mess is quite difficult. (Chakrabarti et al. 1999)

So far people have relied on search engines that hunt for specific words or terms. Text searches frequently retrieve tens of thousands of pages, many of them useless. The problem is how is possible to locate quickly only the information which is needed, and be sure that it is authentic and reliable. (Chakrabarti et al. 1999)

The other approach to find the pages is to use produced lists, which would encourage users to browse the WWW. The production of hierarchical browsing tools has sometimes led to the adoption of library classification schemes to provide the subject hierarchy. (Brümmer et al. 1997a)

1.1 IPMAN- project

Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology started IPMAN-project in January 1999. It is financed by TEKES, Nokia Networks Oy and Open Environment Software Oy. In 1999 the project produced a literary research, which was published in Publications in Telecommunications Software and Multimedia.

The objective of the IPMAN-project is to research increasing Internet Protocol (IP) traffic and it’s affects to the network architecture and the network management. The data volumes will explode in growth in the near future when

new Internet related services enable more customers, more interactions and more data per interaction.

Solving the problems of the continuous growing volumes of Internet is important for the business world as networks and distributed processing systems have become critical success factors. As networks have become larger and more complex, automated network management has come unavoidable in the network management.

In IPMAN-project the network management has been divided into four levels: Network Element Management, Traffic Management, Service Management and Content Management. Levels can be seen in figure 1.

Content Management
Service Management
Traffic Management
Network Element Management

Figure 1. Network management levels in IPMAN-project (Uosukainen et al. 1999, p. 14)

The network element management level is dealing with questions of how to manage network elements in the IP network. The traffic management level is intending to manage the network so that expected traffic properties are achieved. Service management level manages service applications and platforms. The final level is content management and it is dealing with managing the content provided by the service applications.

During the year 1999 the main stress was to study the service management. The aim of the project during the year 2000 is to concentrate to study the content management and the main stress is to create a prototype. The prototype's subject

is content personalization. Content personalization means that a user can influence to the content he wants to get. My task in IPMAN-project is to find out different methods of classification possible to use in IP networks. The decision of the method, which is to be used in the prototype, will be based on my settlement.

1.2 Scope of the thesis

The Web contains approximately 300 million hypertext pages. The amount of pages continues to grow at roughly a million pages per day. The variation of pages is large. The set of Web pages lacks a unifying structure and shows more authoring style and content variation than has seen in traditional text-document collections. (Chakrabarti et al. 1999b, p. 60)

The scope of this thesis is to focus on different classification and indexing methods, which are useful in text classification or indexing in IP networks. Information retrieval is one of the most popular research subjects of today. The main purpose of many study groups is to develop an efficient and useful classification or indexing method to be used for information retrieval in Internet. This thesis will introduce the basic methods of classification and indexing and some of the latest applications and projects where those methods are used. The main purpose is to find out what kind of applications for classification and indexing have been generated lately and the advantages and weaknesses of them. An appropriate method for text classification and indexing will make IP networks, especially Internet, more useful as well to end-users as to content providers.

1.3 Structure of the thesis

In chapter two there is description of metadata and possible ways to use it. In chapter three and four there is described different existing indexing and classification methods.

In chapter five is described how classification and indexing is put into practice in Internet of today. Also the problems and the demands of the future are examined in chapter five. In chapter six is introduced new applications which use existing classification and indexing methods. The purpose has been to find a working and existing application of each method. Anyway, there are also introduced few applications which are just experiments.

Chapter seven includes conclusions of all methods and applications and chapter eight includes the summary. The results of the thesis are reported in eight chapters and the main contents are outlined in figure 2.

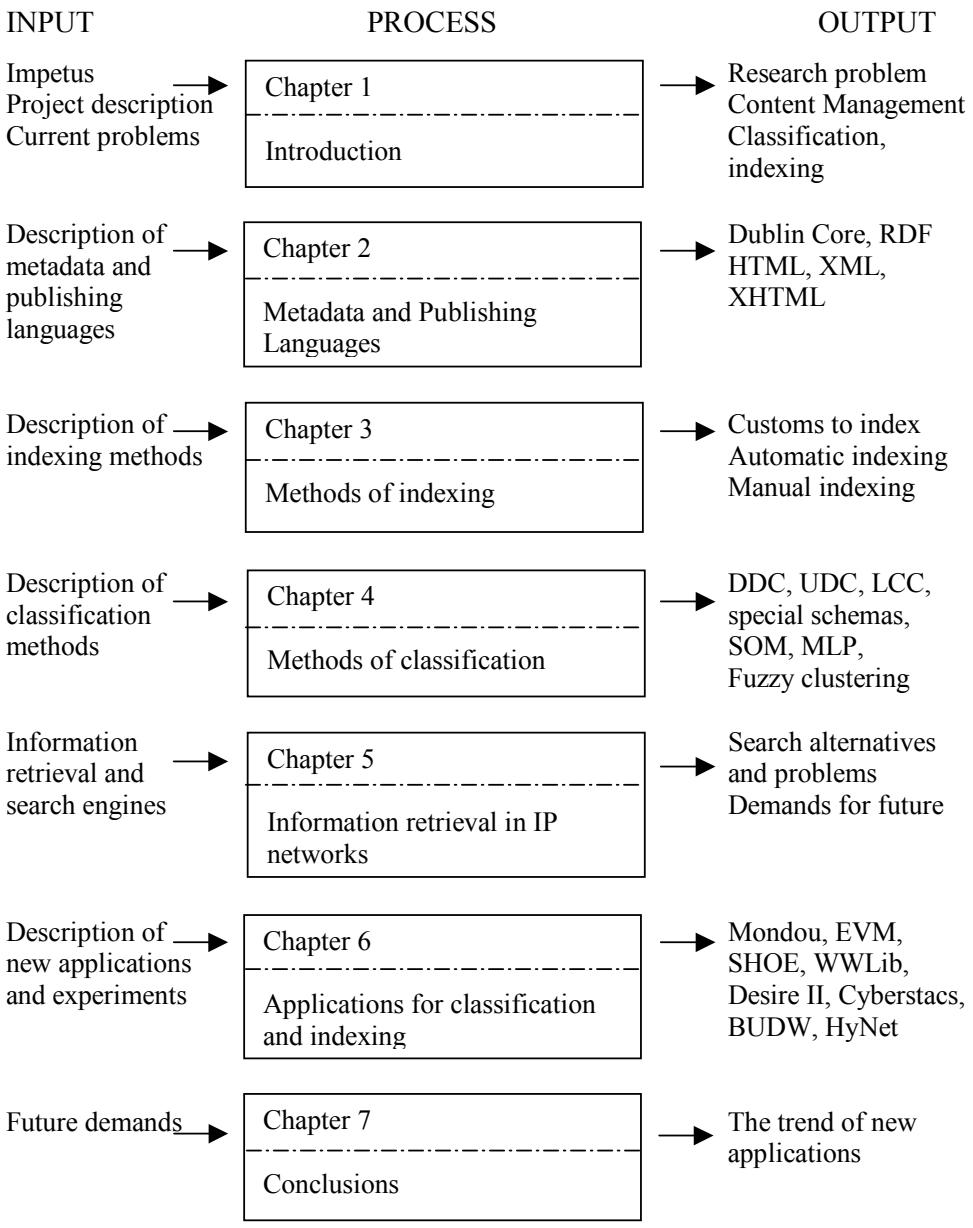

Figure 2. Outline of the thesis.

2 METADATA AND PUBLISHING LANGUAGES

Metadata and publishing languages are explained in this chapter. One way to make classification and indexing easier is to add metadata to an electronic resource situated in network. The metadata that is used in electronic libraries (eLibs) is based on Dublin Core metadata element set. Dublin Core is described in chapter 2.1.1. The eLib metadata uses the 15 Dublin Core attributes. Dublin Core attributes are also used in ordinary web pages to give metadata information to search engines.

Resource Description Framework (RDF) is a new architecture meant for metadata on the Web, especially for diverse metadata needs for separate publishers on the web. It can be used in resource discovery to provide better search engine capabilities and for describing the content and content relationships of a Web page.

Search engines in Internet uses the information embedded in WWW-pages done by some page description and publishing language. In this work, HyperText Markup Language (HTML) and one of the newest languages, Extensible Markup Language (XML), are described after Dublin Core and RDF. Extensible HyperText Markup Language (XHTML) is the latest version of HTML.

XML and XHTML are quite new publishing languages and assumed to attain an important role in publishing in Internet in the near future. Therefore both of them are described more accurately than HTML, which is the main publishing language at present but will apparently make room for XML and XHTML. In chapters of XML and XHTML properties of HTML are brought forward and compared with the properties of XML and XHTML.

2.1 Description of metadata

The International Federation of Library Associations and Institutions gives the following description of metadata:

"Metadata is data about data. The term is used to refer to any data that is used to aid the identification, description and location of networked electronic resources. Many different metadata formats exist, some quite simple in their description, others quite complex and rich." (IFLA 2000)

According to other definition: metadata is machine understandable information about web resources or other things. (Berners-Lee 1997)

The main purpose of metadata is to give some information about the document for computers that cannot deduce this information from the document itself. Keywords and descriptions are supposed to present the main concepts and subjects of the text. (Kirsanov 1997a)

Metadata is open to abuse, but it's still the only technique capable of helping computers for better understanding of human-produced documents. According to Kirsanov, we won't have another choice but to rely on some sort of metadata information until computers achieve a level of intelligence comparable to that of human beings. (Kirsanov 1997a)

Information of metadata consists of a set of elements and attributes, which are needed in description of a document. For instance, the library card indexing is a metadata method. It includes descriptive information like creator, title, the year of publication among others of a book or other document existing in library. (Stenvall and Hakala 1998)

Metadata can be used in documents in two ways:

- the elements of metadata are situated in separated record, for instance in library card index, or
- the elements of metadata are embedded in the document.

(Stenvall and Hakala 1998)

Once created metadata can be interpreted and processed without human assistance, because of its machine-readability. After extracted from the actual content, it should be possible to transfer and process it independently and separately from the original content. This allows the operations only on the metadata instead of the whole content. (Savia et al. 1998)

2.1.1 Dublin Core element set

In March 1995 OCLC/NCSA Metadata Workshop agreed a core list of metadata elements called Dublin Metadata Core Element Set. Dublin Core is shortening for it. Dublin Core provides a standard format (Internet standard RFC2413) for metadata and ensures interoperability for the eLib metadata. The eLib metadata uses the 15 appropriate Dublin Core attributes. (Gardner 1999)

The purpose of Dublin Core metadata element set is to facilitate discovery of electronic resources. It was originally conceived for author-generated description of Web resources but it has also attracted the attention of formal resource description communities such as museums, libraries, government agencies, and commercial organizations. (DCMI 2000c)

Dublin Core is trying to catch several characteristics analyzed below:

Simplicity

- it is meant to be usable for all users, to non-catalogers as well as resource description specialists.

Semantic Interoperability

- the possibility of semantic interoperability across disciplines increases by promoting a commonly understanding set of descriptors that helps to unify other data content standards.

International Consensus

- it is critical to the development of effective discovery infrastructure to recognize the international scope of resource discovery on the Web.

Extensibility

- it provides an economical alternative to more elaborate description models.

Metadata modularity on the Web

- the diversity of metadata needs on the Web requires an infrastructure that supports the coexistence of complementary, independently maintained metadata packages. (DCMI 2000b)

Each Dublin Core element is optional and repeatable. Most of the elements have also specifiers, which make the meaning of the element more accurate. (Stenvall and Hakala 1998)

The elements are given descriptive names. The intention of descriptive names is to make it easier to user to understand the semantic meaning of the element. To promote global interoperability, the element descriptions are associated with a controlled vocabulary for the respective element values. (DCMI 2000a)

Element Descriptions

1. Title

Label: Title

The name given to the resource usually by the creator or publisher.

2. Author or Creator

Label: Creator

The person or organization primarily responsible for creating the intellectual content of the resource.

3. Subject and Keywords

Label: Subject

The topic of the resource. Typically, subject will be expressed as keywords or phrases that describe the subject or content of the resource.

4. Description

Label: Description

A textual description of the content of the resource.

5. Publisher

Label: Publisher

The entity responsible for making the resource available in its present form, like a publishing house, a university department, or a corporate entity.

6. Other Contributor

Label: Contributor

A person or organization that has made significant intellectual contributions to the resource but was not specified in a Creator element.

7. Date

Label: Date

The date the resource have done or been available.

8. Resource Type

Label: Type

The category in which the resource belongs, such as home page, novel, poem, working paper, technical report, essay, dictionary.

9. Format

Label: Format

The data format used to identify the software and sometimes also the hardware that is needed to display or operate the resource. Dimensions, size, duration e.g. are optional and can be also performed in here.

10. Resource Identifier

Label: Identifier

A string or a number is used to identify the resource. Identifier can be for example URLs (Uniform Resource Locator), URNs (Uniform Resource Number) and ISBNs (International Standard Book Number).

11. Source

Label: Source

This contains information about a second resource from which the present resource is derived if it is considered important for discovery of the present resource.

12. Language

Label: Language

The language used in the content of the resource.

13. Relation

Label: Relation

The second resource's identifier and its relationship to the present resource. This element is used to express linkages among related resources.

14. Coverage

Label: Coverage

The spatial and/or temporal characteristics of the intellectual content of the resource. Spatial coverage refers to a physical region. Temporal coverage refers to the content of the resource.

15. Rights Management

Label: Rights

An identifier that links to a rights management statement, or an identifier that links to a service providing information about rights management for the resource. (Weibel et al. 1998)

2.1.2 Resource Description Framework

The World Wide Web Consortium (W3C) has begun to implement an architecture for metadata for the Web. The Resource Description Framework (RDF) is designed with an eye to many diverse metadata needs of vendors and information providers. (DCMI 2000c)

RDF is meant to support the interoperability of metadata. It allows any kind of Web resources, in other words, any object with a Uniform Resource Identifier (URI) as its address, to be made available in machine understandable form. (Iannella 1999)

RDF is meant to be metadata for any object that can be found on the Web. It is a means for developing tools and applications using a common syntax for describing Web resources. In the year 1997 the W3C recognized the need for a language, which would eliminate the problems of content ratings, intellectual property rights and digital signatures while allowing all kinds of Web resources to be visible and be discovered in the Web. A working group within the W3C has drawn up a data model and syntax for RDF. (Heery 1998)

RDF is designed specifically with the Web in mind, so it takes into account the features of Web resources. It is a syntax based on a data model, which influences the way properties are described. The structure of descriptions is explicit and means that RDF has a good fit for describing Web resources. From another direction, it might cause problems within environments where there is a need to re-use or interoperate with 'legacy metadata' which may well contain logical inconsistencies. (Heery 1998)

The model for representing properties and property values is the foundation of RDF and the basic data model consists of three object types:

Resources:

Resources can be called all things described by RDF expressions. A resource can be an entire Web page, like an HTML document or a part of a Web page like an element within the HTML or XML document source. A resource may also be a whole collection of pages, like an entire Web site. An object that is not directly accessible via the Web, like a printed book, can also be considered as a resource. A resource will always have URI and an optional anchor Id.

Properties:

A resource can be described as a used property that can have a specific aspect, characteristic, attribute or relation. Each property has a specific meaning, and it defines its permitted values, the types of resources it can describe, and its relationship with other properties.

Statements:

A RDF statement is a specific resource together with a named property plus the value of that property for that resource. These three parts of a statement are called the subject, the predicate, and the object. The object of a statement can be another resource or it can be a literal. This means a resource specified by an URI or a simple string or other primitive data type defined by XML. (Lassila and Swick 1999)

The following sentences can be considered as an example:

The individual referred to by employee id 92758 is named Kirsi Lehtinen and has the email address klehtine@lut.fi. The resource http://www.lut.fi/~klehtine/index.html was created by this individual.

The sentence is illustrated in figure 3.

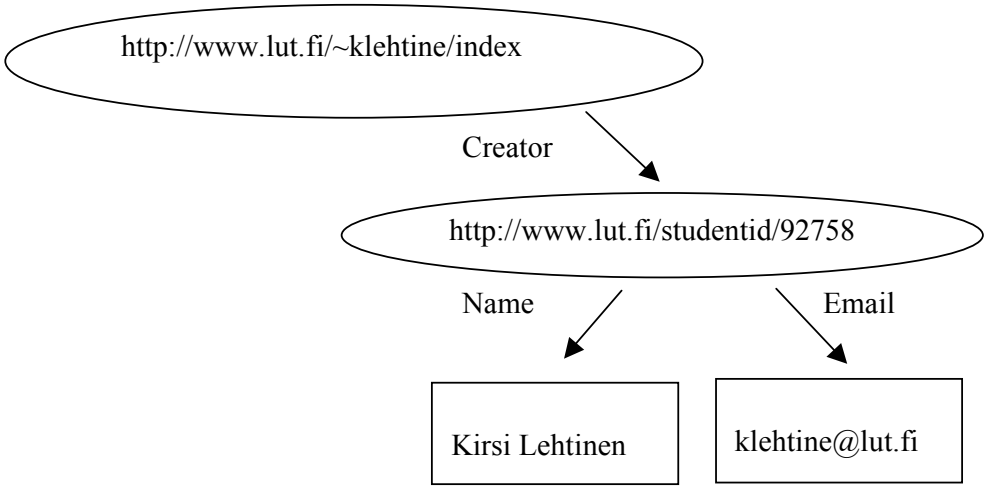

Figure 3. RDF property with structured value. (Lassila and Swick 1999)

The example is written in RDF/XML in the following way:

```
<rdf:RDF>
  <rdf:Description about="http://www.lut.fi/~klehtine/index">
    <s:Creator rdf:resource="http://www.lut.fi/studentid/92758"/>
  </rdf:Description>
  <rdf:Description about="http: ://www.lut.fi/studentid/92758"/>
    <v:Name>Kirsi Lehtinen</v:Name>
    <v:Email>klehtine@lut.fi</v:Email>
  </rdf:Description>
</rdf:RDF>  (Lassila and Swick 1999)
```

2.2 Description of publishing languages

A universally understood language is needed for publishing information globally. It should be a language that all computers may potentially understand. (Raggett 1999) The most famous and common language, for page description and publishing on the Web is HyperText Markup Language (HTML). It describes the contents and appearance of the documents publishing on the Web. Publishing languages are formed from entities, elements and attributes. Because HTML has become insufficient for the needs of publication other languages have developed. Extensible Markup Language (XML) has developed to be a language, which better satisfy the needs of information retrieval and diverse browsing devices. Its purpose is to describe the structure of the document without responding the appearance of the document. Extensible HyperText Markup Language (XHTML) is a combination of HTML and XML.

2.2.1 HyperText Markup Language

HyperText Markup Language (HTML) was originally developed by Tim Berners-Lee while he was working at CERN. NCSA developed the Mosaic

browser, which popularized HTML. During the 1990s it has been a success with the explosive growth of the Web. Since beginning, HTML has been extended in number of ways. (Raggett 1999)

HTML is a universally understood publishing language used by the WWW. (Raggett 1999) Information of metadata can be embedded in HTML document. With the help of metadata an HTML document can be classified and indexed.

Below are listed properties of HTML:

- Online documents can include headings, text, tables, lists, photos, etc.
- Online information can be retrieved via hypertext links just by clicking a button.
- Forms for conducting transactions with remote services can be designed like for use in searching for information, making reservations, ordering products, etc.
- Spreadsheets, video clips, sound clips, and other applications can be included directly in documents. (Raggett 1999)

HTML is a non-proprietary format based upon Standard General Markup Language (SGML). It can be created and processed by a wide range of tools, from simple plain text editors to more sophisticated tools. To structure text into headings, paragraphs, lists, hypertext links etc., HTML uses tags such as <h1> and </h1>. (Raggett et al. 2000)

A typical example of HTML code could be as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>My first HTML document</TITLE>
  </HEAD>
```

```
  <BODY>
    <P>Hello world!
  </BODY>
</HTML>
```
(W3C HTML working group 1999)

2.2.2 Extensible Markup Language

The Extensible Markup Language (XML) is a subset of SGML. (Bray et al. 1998). XML is a method developed for putting structured data in a text file whereby it can be classified and indexed.

XML allows to define one's own markup formats. XML is a set of rules for designing text formats for data, in a way that produces files that are easy to generate and read especially by a computer. That produced data is often stored on a disk in binary format or text format. Text format allows, when needed, to look at the data without the program that produced it. (Bos 2000)

The rules for XML files are much stricter than for HTML. It means that a forgotten tag or an attribute without quotes makes the file unusable, while in HTML such practice is at least tolerated. According to XML specification applications are not allowed to try to show a broken XML file. If the file is broken, an application has to stop and issue an error. (Bos 2000)

The design goals for XML have been diverse. To be usable over the Internet, to support a wide variety of applications and to be compatible with SGML can be regarded as most important design goals. Writing programs, which process XML documents, with ease and to minimize the number of optional features in XML have also been some of the design goals for XML. Other goals have been that XML documents should be legible and reasonably clear, the preparation of XML design should be quick and the design of XML shall be formal. XML documents shall also be easy to create. (Bray et al. 1998)

In each XML document is a logical and a physical structure. Physically, the document is composed of entities, which can also called objects. A document begins in a "root" by declaration of the XML-version like: <?xml version=”1.0”?>. Logically, the document is composed among other things of declarations, elements, comments, character references, and other possible things indicated in the document by explicit markup. The logical and physical structures must nest properly. (Bray et al. 1998)

The XML document is composed of different *entities*. There can be one or more logical *elements* in each entity. Each of these elements can have certain *attributes* (properties) that describe the way in which it is to be processed. The relationships between the entities, elements and attributes are described in a formal syntax of XML. This formal syntax can be used to tell the computer how to recognize to different component parts of a document. (Bryan 1998)

XML uses *tags* and *attributes*, like HTML, but doesn’t specify the meaning of each tag & attribute. XML uses the tags to delimit the structure of the data, and leaves the interpretation of the data completely to the application that reads it. If you see "<p>" in an XML file, it doesn’t necessarily mean a paragraph. (Bos 2000)

A typical example of XML code could be as follows:

```
<memo>from>Martin Bryan</from>
<date>5th November</date>
<subject>Cats and Dogs</subject>
<text>Please remember to keep all cats and dogs indoors tonight.
</text>
</memo>
```

Because the start and the end of each logical element of the file has been clearly identified by entry of a start-tag (e.g. <to>) and an end-tag (e.g. </to> is the form of the file ideal for a computer to follow and to process. (Bryan 1998)

Nothing is said about the format of the final document in the code. That makes it possible to users for example to print the text onto a pre-printed form, or to generate a completely new form where each element of the document has put in new order. (Bryan 1998)

To define tag sets of their own, users must create a *Document Type Definition* (DTD). DTD identifies the relationships between the various elements that form their documents. The XML DTD of previous example of XML code might be according to Bryan (1998) like below:

```
<!DOCTYPE memo [
<!ELEMENT memo    (to, from, date, subject?, para+) >
<!ELEMENT para    (#PCDATA) >
<!ELEMENT to      (#PCDATA) >
<!ELEMENT from    (#PCDATA) >
<!ELEMENT date    (#PCDATA) >
<!ELEMENT subject (#PCDATA) >
]>
```

This DTD tells the computer that a memo consists of next header elements: <to>, <from> and <date>. Header element <subject> is optional, which must be followed by the contents of the memo. The contents of the memo defined in this simple example is made up of a number of paragraphs, at least one of which must be present (this is indicated by the + immediately after `para`). In this simplified example a paragraph has been defined as a leaf node that can contain *parsed character data* (`#PCDATA`), i.e. data that has been checked to ensure that it contains no unrecognized markup strings. In a similar way the `<to>`, `<from>`, `<date>` and `<subject>` elements have been declared to be leaf nodes in the document structure tree. (Bryan 1998)

XML-documents are classified in two categories: *well-formed* and *valid*. A well-formed document is done according to XML definition and syntax. Also detailed

conditions have set to the attributes and entities in XML documents. (Walsh 1998)

In XML it is not possible to exclude specific elements from being contained within an element like in SGML. For example, in HTML 4, strict DTD forbids the nesting of an 'a' element within another 'a' element to any descendant depth. It is not possible to spell out these kind of prohibitions in XML. Even though these prohibitions cannot be defined in the DTD, there are certain elements that should not be nested. A summary of such elements and the elements that should not be nested in them is found normative in XHTML 1.0 specification. (W3C HTML working group 2000)

A XML document is well-formed if it meets all the well-formedness constraints given in XML 1.0 specification. Also each of the parsed entities which is referenced directly or indirectly within the document should be well-formed. (Bray et al. 1998)

A XML document is valid if it has an associated document type declaration and if the document complies with the constraints expressed in it. The document type declaration should be before the first element in the document and contain or point to markup declarations that provide a grammar for a class of documents. This grammar is known as a document type definition, or DTD. The document type declaration can point to an external subset containing markup declarations, or can contain the markup declarations directly in an internal subset, or can do both. (Bray et al. 1998)

XML is defined by specifications described below:

- **XML**, the Extensible Markup Language
Defines the syntax of XML.

- **XSL**, the Extensible Stylesheet Language

Expresses the stylesheets and consists of two parts:

- a language for transforming XML documents, and
- an XML vocabulary for specifying formatting semantics.

An XSL stylesheet specifies how transforming into a XML document that uses the formatting vocabulary shall be done. (Lilley and Quint 2000)

- **XLink**, the Extensible Linking Language

Defines how to represent links between resources. In addition to simple links, Xlink allows elements to be inserted into XML documents in order to create and describe links between multiple resources and links between read-only resources. It uses XML syntax to create structures that can describe the simple unidirectional hyperlinks, as well as more sophisticated links. (Connolly 2000)

- **XPointer,** the Extensible Pointer Language

The XML Pointer Language (XPointer) is a language to be used as a fragment identifier for any URI-reference (Uniform Resource Identifier) that locates a resource of Internet media type text/xml or application/xml. (Connolly 2000)

2.2.3 Extensible HyperText Markup Language

Extensible HyperText Markup Language (XHTML) 1.0 is W3C's recommendation for the latest version of HTML, succeeding earlier versions of HTML. XHTML 1.0 is a reformulation of HTML 4.01, and is meant to combine the strength of HTML 4 with the power of XML. (Raggett et al. 2000).

XHTML 1.0 reformulates the three HTML 4 document types as an XML application, which makes it easier to process and easier to maintain. XHTML 1.0 have tags like in HTML 4 and is intended to be used as a language for content that is both XML-conforming but can also be interpreted by existing browsers, by following a few simple guidelines. (Raggett et al. 2000)

According to W3C, there are following benefits in XHTML for developers:

- XHTML documents are XML conforming and therefore readily viewed, edited, and validated with standard XML tools.
- XHTML documents can be written to operate as well in existing HTML 4 conforming user agents, as in new, XHTML 1.0 conforming user agents.
- XHTML documents can utilize applications (e.g. scripts and applets) that rely upon either the HTML Document Object Model or the XML Document Object Model.
- As the XHTML family evolves, documents conforming to XHTML 1.0 will likely interoperate within and among various XHTML environments. (W3C HTML working group 2000)

Content developers can remain confident in their content's backward and future compatibility in entering the XML word by migrating to XHTML, and in that way get all attendant benefits of XML. (W3C HTML working group 2000)

Some of the benefits of migrating to XHTML are described above:

- In XML, it is quite easy to introduce new elements or additional element attributes for new ideas. The XHTML family is designed to accommodate these extensions through XHTML modules and techniques for developing new XHTML-conforming modules. These modules will permit the combination of existing and new feature sets when developing content and when designing new user agents.
- Internet document viewing will be carried out on alternate platforms and therefore the XHTML family is designed with general user agent interoperability in mind. Through a new user agent and document profiling mechanism, servers, proxies, and user agents will be able to perform best effort content transformation. By XHTML it will be

> possible to develop a content that is usable by any XHTML-conforming user agent. (W3C HTML working group 2000)

Because the use of XHTML makes other platforms than traditional desktops possible to use, all of the XHTML elements will not be required on all used platforms. This means, for example, that a hand held device or a cell-phone may only support a subset of XHTML elements. (W3C HTML working group 2000)

A strictly conforming XHTML document must meet all of the following criteria:

1. It must validate against one of the three DTD modules:
   a) DTD/xhtml1-strict.dtd, which is identified by the PUBLIC and SYSTEM identifiers:
   - PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   - SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"

   b) DTD/xhtml1-transitional.dtd, which is identified by the PUBLIC and SYSTEM identifiers:
   - PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   - SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"

   c) DTD/xhtml1-frameset.dtd, which is identified by the PUBLIC and SYSTEM identifiers:
   - PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
   - SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd"

The Strict DTD is used normally, but when support for presentation attribute and elements are required, the Transitional DTD should be used. Frameset DTD should be used for documents with frames.

2. The root element of the document must be <html>.
3. The root element of the document must use the xmlns attribute and the namespace for XHTML is defined to be http://www.w3.org/1999/xhtml.

4. There must be a DOCTYPE declaration in the document before the root element. The public identifier included in the DOCTYPE declaration must reference one of the three DTDs (mentioned in item number 1) using the respective formal public identifier. The system identifier may be changed to reflect local system conventions. (W3C HTML working group 2000)

Here is an example according to W3C HTML working group (2000) of a minimal XHTML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Virtual Library</title>
</head>
<body>
<p>Moved to <a href="http://vlib.org/">vlib.org</a>.</p>
</body>
</html>
```

XML declaration is included in example above and is not required in all XML documents. XML declarations are required when the character encoding of the document is other than the default UTF-8 or UTF-16. (W3C HTML working group 2000)

Because XHTML is an XML application, certain legal practices in SGML-based HTML 4 must be changed. According to XML and its well-formedness all elements must either have closing tags or be written in a special form, and all the elements must nest. XHTML documents must use lower case for all HTML element and attribute names, because in XML e.g. and <LI> are different tags. (W3C HTML working group 2000)

XHTML 1.0 provides the basis for a family of document types that will extend and make subsets in XHTML, to support a wide range of new devices and applications. This is possible by defining modules and specifying a mechanism for combining these modules. This mechanism will enable the extension and sub-setting of XHTML in a uniform way through the definition of new modules. (W3C HTML working group 2000)

Modularization breaks XHTML up into a series of smaller element sets. These elements can then be recombined and in that way to be usable to the needs of different communities. (W3C HTML working group 2000)

Modularization brings with it several advantages:

- a formal mechanism for sub-setting XHTML.
- a formal mechanism for extending XHTML.
- a transformation between document types is simpler.
- the reuse of modules in new document types.
  (W3C HTML working group 2000)

The syntax and semantics of a set of documents is specified in a document profile. The document profile specifies the facilities required to process different types of documents, for example which image formats can be used, levels of scripting, style sheet support, and so on. Conformance to a document profile is a basis for interoperability. (W3C HTML working group 2000)

This enables product designers to define their own standard profiles. For different clients there is no need to write several different versions of documents. Also for special groups such as chemists, medical doctors, or mathematicians this allows a special profile to be built using standard HTML elements and a group of elements dedicated especially to the specialist's needs. (W3C HTML working group 2000)

3 METHODS OF INDEXING

This chapter introduces indexing and the most common methods of it. The objective of indexing is to transform the received items to the searchable data structure. All data that search systems use are indexed some how. Also hierarchical classification systems required indexed databases to their operations. Indexing can be carried out automatically as well as manually and the last subchapter handles this subject. Indexing is originally called cataloging.

3.1 Description of indexing

Indexing is a process of developing a document representation by assigning content descriptors or terms to the document. These terms are used in assessing the relevance of a document to a user query and directly contribute to the retrieval effectiveness of an information retrieval (IR) system. There are two types of terms: objective and non-objective. In general there is no disagreement about how to assign objective terms in the document. Objective terms are applying integrally to the document. Author name, document URL, and date of publication are examples of objective terms. In contrast, there is no agreement about the choice or the degree of applicability of non-objective terms to the document. These are intended to relate to the information content that is manifested in the document. (Gudivara et al. 1997)

However the search-engines which offer the information to the users always require some kind of indexing system. The way in which such search-engines assemble their data can vary from simple**,** based on straight-forward text string matching of document content to complex, involving the use of factors such as:

- relevance weighting of terms, based on some combination of frequency, and (for multiple search terms) proximity

- occurrence of words in the first *n* words of the document extraction of keywords (including from META elements, if present). (Wallis and Burden 1995)

3.2 Customs to index

This chapter introduces the most common ways to index documents in the Web, which are: full-text indexing, inverted indexing, semantic indexing and latent semantic indexing.

3.2.1 Full-text indexing

Full-text indexing means that every keyword from a textual document appears in the index. Because this is a method that can be automated it is therefore desirable for a computerized system. There are algorithms to reduce the number of indexed less relevant terms by identifying and ignoring them. In these algorithms, the weighting is often determined by the relationship between the frequency of the keyword in the document and its frequency in the documents as a whole. (Patterson 1997)

3.2.2 Inverted indexing

Inverted index is an index of all terms of keywords that occur in all documents. Each keyword is stored with a list of all documents that contain the keyword. This method requires huge amounts of processing to maintain. The number of keywords stored in the index could be reduced using the algorithms mentioned for full-text indexing, but it still requires a large amount of processing and storage space. (Patterson 1997)

3.2.3 Semantic indexing

Semantic indexing is based on the characteristics of different file types and this information is used in indexing. Semantic indexing requires firstly that the file type is identified and secondly, that an appropriate indexing procedure is adopted according to the field type identified. This method can extract information from files other than purely text files, and can decide where high-quality information is to be found and retrieved. This leads to comprehensive but smaller indexes. (Patterson 1997)

3.2.4 Latent semantic indexing

The latent semantic structure analysis needs more than a keyword alone for indexing. In each document each keyword and the frequency of each keyword must be stored. The document matrix is to be formed with the help of the stored frequency and keywords. The document matrix is used as input to latent semantic indexing. There a single valued decomposition is applied to the document matrix to obtain 3 matrices, one of which corresponds to a number of dimensions of vectors for the terms, and one other to the number of dimensions of vectors to the documents. These dimensions can be reduced to 2 or 3 and used to plot co-ordinates in 2 or 3 dimensional space respectively. (Patterson 1997)

3.3 Automatic Indexing vs. manual indexing

Indexing can be carried out either manually or automatically. Trained indexers or human experts in the subject area of the document perform manual indexing. Manual indexing is made by using a controlled available vocabulary in the form of terminology lists. Also the indexers and experts follow the instructions for the use of the terms. Because of the size of the Web and the diversity of subject material present in Web documents, manual indexing is not practical. Automatic indexing relies on a less tightly controlled vocabulary and entails many more

aspects in representing of a document than is possible under manual indexing. This helps to retrieve a document to a great diversity of user queries. (Gudivara et al. 1997)

In human indexing the advantages are the ability to determine concept abstraction and judge the value of a concept and the disadvantages over automatic indexing are cost, processing time and consistency. After the initial hardware cost is amortized, the costs of automatic indexing are as part of the normal operations and maintenance costs of the computer system. There are no additional indexing costs like the salaries and other benefits to pay to human indexers. (Kowalski 1997, p. 55-56)

Also according to Lynch, 1997, automating information access has the advantage of directly exploiting the rapidly dropping costs of computers and avoiding the high expense and delays of human indexing.

Another advantage to automatic indexing is the predictability of the behavior of the algorithms. If the indexing is being performed automatically by an algorithm, there is consistency in the index term selection process. Human indexers generate different indexing for the same document. (Kowalski 1997, p. 56)

The strength in manual indexing is the human ability to consolidate many similar ideas into a small number of representative index terms. Automated indexing systems try to achieve these by using weighted and natural language systems and by concept indexing. (Kowalski 1997, s. 63)

An experienced researcher understands the automatic indexing process and is able to predict its utilities and deficiencies, trying to compensate or utilize the system characteristics in a search strategy. (Kowalski 1997, s. 56)

In automatic indexing the system is capable to automatically determine the index terms to be assigned to an item. If the intention is to emulate a human indexer and determine a limited number of index terms for the major concepts in the item a full-text indexing is not enough but more complex processing is required (Kowalski 1997, s. 54)

4 METHODS OF CLASSIFICATION

The aim of this chapter is to explain diverse classification methods in general. First are explained classification methods used in virtual-libraries like Dewey Decimal Classification (DDC), Universal Decimal Classification (UDC), Library of Congress Classification (LCC) and some other methods. The same methods are used in conventional libraries. Then comes mathematical methods like soft computing systems: Self-Organizing Map (SOM / WEBSOM), Multi-Layer Perceptron Network (MLP) and fuzzy systems. Also other classification systems exist, but are not explained in this thesis. One of them is for example a statistical nearest neighbor-method. However, the methods explained here are the most common and utilized in textual indexing and classification systems.

4.1 Description of classification

Classification has defined by Chen et al. in 1996 as follows: "Data classification is the process which finds the common properties among a set of objects in a database and classifies them into different classes, according to a classification model."

There are several different types of classification systems around, varying in scope, methodology and other characteristics. (Brümmer et al. 1997a)

Below are some customs listed about classification systems:

- by subject coverage: general or subject specific
- by language: multilingual or individual language
- by geography: global or national
- by creating/supporting body: representative of a long-term committed body or a homegrown system developed by a couple of individuals

- by user environment: libraries with container publications or documentation services carrying small focused documents (e.g. abstract and index databases)
- by structure: enumerative or faceted
- by methodology: a priori construction according to a general structure of knowledge and scientific disciplines or using existing classified documents. (Brümmer et al. 1997a)

The types mentioned above show what types of classification scheme are theoretically possible. In reality, the most frequently used types of classification schemes are:

- universal,
- national general,
- subject specific schemes, most often international,
- home-grown systems, and
- local adaptations of all types. (Brümmer et al. 1997a)

Under 'universal' schemes is included schemes which are global geographically and multilingual in scope and aim to include all possible subjects.
(Brümmer et al. 1997a)

Subsequently, here are some advantages for classified Web-knowledge:

- Able to be browsed easily.
- Searches can be broadening and narrowing.
- Gives a context to the used search terms.
- Potential to permit multilingual access to a collection.
- Classified lists can be divided into smaller parts if required.
- The use of an agreed classification scheme could enable improved browsing and subject searching across databases.

- An established classification system is not usually in danger of obsolescence.
- They have the potential to be well known, because of regular users of libraries is familiar with at least some traditional library scheme.
- Many classification schemes are available in machine-readable form. (Brümmer et al. 1997a)

4.2 Classification used in libraries

The most widely used classification schemes in universal are Dewey Decimal Classification (DDC), the Universal Decimal Classification (UDC) and the classification scheme devised by the Library of Congress Classification (LCC). Classification schemes mentioned above were developed for the use of libraries since the late nineteenth century. (Brümmer et al. 1997a)

4.2.1 Dewey Decimal Classification

The Dewey Decimal Classification System (DDC) was originally being produced in 1876 for a small North American College library by Melvil Dewey. DDC is distributed in Machine-Readable Cataloguing (MARC) records produced by the Library of Congress (LC) and some bibliographic utilities. (Brümmer et al. 1997b)

The DDC is the most widely used hierarchical classification scheme in the world. Numbers represent a concept and each concept and its position in the hierarchy can be identified by the number. (Patterson 1997) DDC system is seen in appendix 1.

4.2.2 Universal Decimal Classification

The Universal Decimal Classification (UDC) was developed in 1895, directly from the DDC by two Belgians, Paul Otlet and Henri LaFontaine. Their task was to create a bibliography of everything that had appeared in print. Mr. Otlet and Mr. LaFontaine extend a number of synthetic devices and add additional auxiliary tables to UDC. (McIlwaine 1998)

The UDC is more flexible than the DDC, and lacks uniformity across libraries that use it. It's not used much in North America but it's used in special libraries, in mathematics libraries, and in science and technology libraries in other English-speaking parts of the world. It is also used extensively in Eastern Europe, South America and Spain. The French National Bibliography basis is from the UDC and it's still used for the National Bibliography in French-speaking Africa. It is also required in all science and technology libraries in Russia. (McIlwaine 1998)

To use UDC classification correctly, the classifier must know the principles of classification well because there is no citation orders laid down. An institution must decide on its own rules and maintain its own authority file. (McIlwaine 1998)

4.2.3 Library of Congress Classification

The Library of Congress Classification System (LCC) is one of the world's most widely spread classification schemes. Two Congress Librarians, Dr. Herbert Putnam and his Chief Cataloguer Charles Martel decided to start a new classification system for the collections of the Library of Congress in 1899. Basic features were taken from Charles Ammi Cutter's Expansive Classification. (UKOLN Metadata Group 1997)

Putnam built LCC as an enumerative system which has 21 major classes, each class being given an arbitrary capital letter between A-Z, with 5 exceptions: I, O, W, X, Y. After this Putnam delegated the further development to specialists, cataloguers and classifiers. The system was and still is decentralized. The different classes and subclasses were published for the first time between 1899-1940. This has lead to the fact that schedules often differ very much in number and the kinds of revisions accomplished. (UKOLN Metadata Group 1997) LCC system is seen in appendix 2.

4.2.4 National general schemes

Most of the advantages and disadvantages of universal classification schemes apply also to national general schemes. National general schemes have also additional characteristics that make them perhaps not the best choice for an Internet service. An Internet service claims to be relevant for a wider user group than one limited to certain national boundaries. (Brümmer et al. 1997a)

4.2.5 Subject specific and home-grown schemes

Many special subject specific schemes have been devised for a particular user-group. Typically they have been developed for use with indexing and abstracting services, special collections or important journals and bibliographies in a scientific discipline. They do have the potential to provide a structure and terminology much closer to the discipline and can be brought up to date easily, compared to universal schemes. (Brümmer et al. 1997a)

Some Web sites like Yahoo! have tried to organize knowledge on the Internet by devising the classification schemes of their own. Yahoo! lists Web sites using it's own universal classification scheme which contains 14 main categories. (Brümmer et al. 1997a)

4.3 Neural network methods and fuzzy systems

Neural computing is a branch of computing whose origins date back to the early 1940s. Conventional computing has overshadowed the neural computing, but advances in computer hardware technology and the discovery of new techniques and developments has led it to new popularity in the late 1980s. (Department of Trade and Industry 1993, p. 2.1)

Neural networks have following characteristics. They can:

- learn from experience,
- generalize from examples, and
- abstract essential information from noisy data. (Department of Trade and Industry 1994, p. 13)

Neural networks can provide good results in short time scale for certain types of problem in short time scales. This is possible only when a great deal of care is taken over neural network design and input data pre-processing design. (Department of Trade and Industry 1994, p. 13)

Among other things there are many attributes to be used as benefits in applications of neural computing systems. Some of the attributes are listed below:

- Learning from experience: neural networks are suited to problems provided a large amount of data from which a response can be learnt and whose solution is complex and difficult to specify.
- Generalizing from examples: ability to interpolate from previous learning is an important attribute for any self-learning system. Designing carefully is in key position to achieve high levels of generalization and give the correct response to data that it has not previously encountered.
- Extracting essential information from noisy data: because neural networks are essentially statistical systems, they can recognize patterns

underlying process noise and be able to extract information from a large number of examples.

- Developing solutions faster, and with less reliance on domain expertise: neural networks learn by example, and as long as examples are available and an appropriate design is adopted, effective solutions can be constructed quicker than by using traditional approaches.
- Adaptability: the nature of neural networks allows them to learn continuously from new, before unused data, and solutions can be designed to adapt to their operating environment.
- Computational efficiency: training a neural network demands a lot of computer power, but the computational requirements of a fully trained neural network when it is used in recall mode can be modest.
- Non-linearity: neural networks are large non-linear processors whereas many other processing techniques are based on assumptions about linearity, which limit their application to real world problems. (Department of Trade and Industry 1994, pp. 13-14)

Key elements: neuron and network

Neural Computing consists of two key elements: the neuron and the network. Neurons are also called units. These units are connected together into a neural network. (Department of Trade and Industry 1993, p. 2.1) Conceptually, units operate in parallel. (Department of Trade and Industry 1994, p. 15)

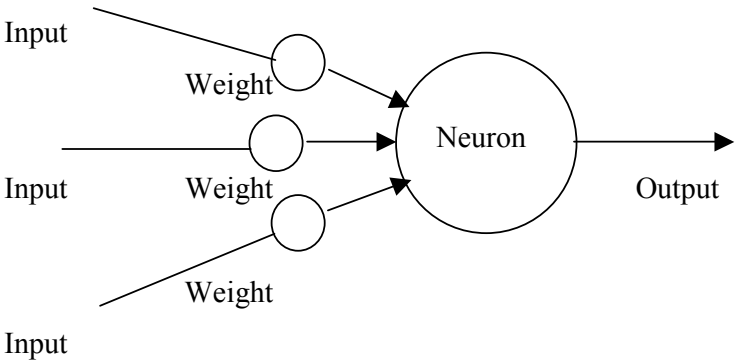

Figure 4. The structure and function of a neuron (Department of Trade and Industry 1993, p. 2.2)

Each neuron within the network takes one or more inputs and produces an output. At each neuron every input has a weight, which modifies the strength of each input connected to that neuron. The neuron adds together all the inputs and calculates an output to be passed on. The structure and function of a neuron is illustrated in figure 4. (Department of Trade and Industry 1993, p. 2.1)

In neural network can be tens to thousands of neurons. Network topology is the way in which the neurons are organized. In figure 5, a network topology is shown which can also called neural network architecture. Neural network architecture consists of layers of neurons. The output of each neuron is connected to all in the next layer. Data flows into the network through the input layer, passes through one or more intermediate hidden layers, and finally flows out of the network through the output layer. The outputs of hidden layers are internal to the network and therefore called hidden. (Department of Trade and Industry 1993, p. 2.2)

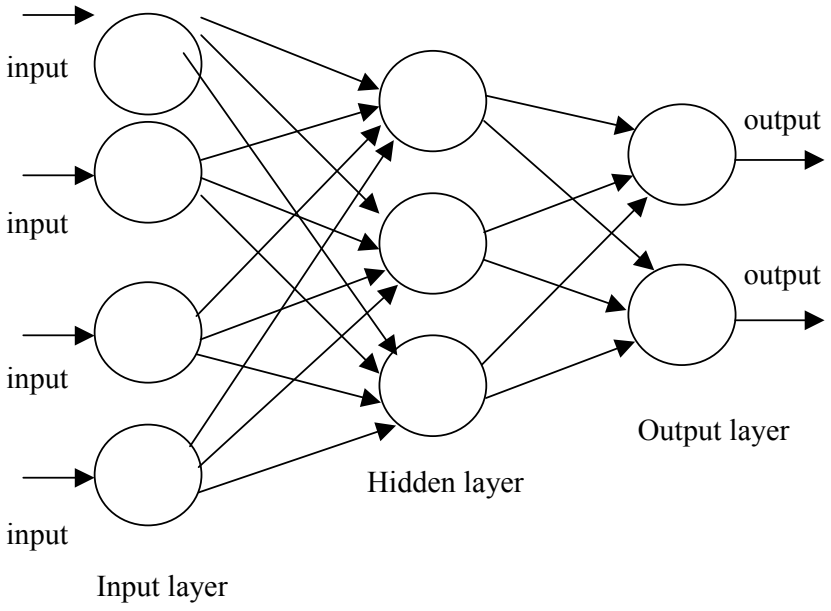

Figure 5. A neural network architecture (Department of Trade and Industry 1993, p. 2.3, Department of Trade and Industry 1994, p.17)

There are many kinds of network topologies and figure 5 shows just one kind of them. In some networks backward as well as forward connections are possible. Some networks allow connections between layers but connections between neurons in the same layer, are impossible and in some networks a neuron can even be connected back to itself. In principle, there can be any number of neurons connected together in any number of layers. (Department of Trade and Industry 1993, p. 2.3)

Neural network weights

The neural network can implement any transformation between its inputs and outputs by varying the weights associated with each input. These weights need to be computed for each particular application. It is not usually possible to compute the weights directly and the process that is needed for getting the right weights is called neural network training. Neural network training is a repetitive and often time-consuming process. (Department of Trade and Industry 1994, s. 15)

Each example is presented to the network during the training process, and the values of the weights are adjusted to bring the output of the whole network closer to that desired. The training phase will end when the network provides sufficiently accurate answers to all the training problems. (Department of Trade and Industry 1993, p. 2.2)

Activation function

Known as a "soft limiting" function, the non-linear "activation function" is the preferred form for most types of neural network. (Figure 6). The non-linearity is critical to the flexibility of the neural network and without this non-linearity, the neural network is limited to implementing simple linear functions like addition and multiplication, between neural network inputs and outputs. With the non-linear unit, the neural network can implement more complex transformations and solve a wide range of problems. (Department of Trade and Industry 1994, p. 15)

Some neural network designs employ a linear "hard limiting" function, which is simpler to compute but may result in difficulties in training the neural network. (Department of Trade and Industry 1994, p. 15)

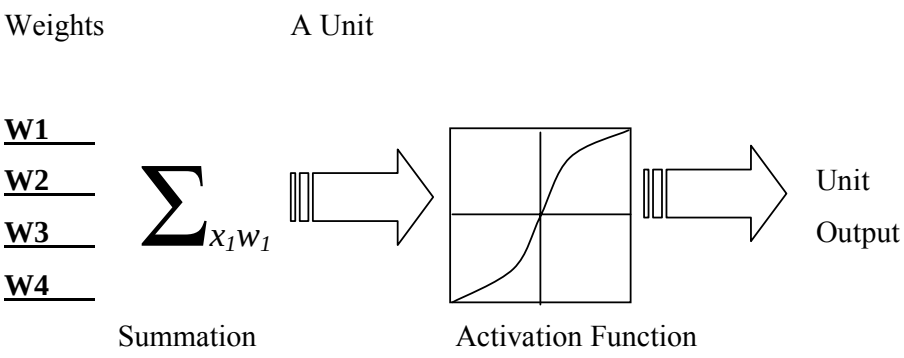

Figure 6. The computation involved in an example neural network unit. (Department of Trade and Industry 1994, p. 15)

Training a neural network

Neural computing does not require an explicit description of how the problem is to be solved. Neural networks develop their own solutions to problems, which means that neural networks are trained rather than programmed. (Department of Trade and Industry 1994, p. 16) The neural network is adapting itself during a training period. This adaptation is based on examples of similar problems with a desired solution to each problem. After sufficient training the neural network is able to relate the problem data to the solutions and it is then able to offer a viable solution to a brand new problem. (Department of Trade and Industry 1994, p. 2.1)

A single neuron has a very limited capability as a device for solving problems but when several neurons are connected together and form a large network, it is possible for the network to be trained to offer a meaningful solution. (Department of Trade and Industry 1994, p. 2.3)

There are many techniques called training algorithms for training a neural network but in general there are two ways to train a neural network with two basic classes of training algorithms. One is called "*supervised training*" and the other is "*unsupervised training*". In supervised training, the neural network adjusts its weights so that its outputs are as near as possible to the given target outputs. MLP is a supervised method. In unsupervised learning the target has not been given to the neural network. Instead, the network learns to recognize any patterns inherent in the data. SOM is an unsupervised method. (Department of Trade and Industry 1994, p. 16)

After being trained, the neural network weights are fixed and the network can be used to predict the solution for some new, previously unseen data. It is also possible to provide a neural network with the ability to continue the training process while it is in use, which means that it is able to adapt itself to the incoming data. Neural networks have this powerful feature that can provide significant advantages over conventional computing systems. (Department of Trade and Industry 1994, p. 17)

Pre- and post-processing

Before the inputs (signals or data) are suitable for the neural network they must convert to the numerical values. This is called pre-processing. These numerical values range from 0 to 1 or -1 to +1, depending upon the type of neural network used. (Department of Trade and Industry 1994, p. 84) In a similar way to convert the neural network outputs from the numeric representation used inside a neural network into a suitable form is called post-processing. (Department of Trade and Industry 1994, p. 17)

Generalization

A trained neural network application is able to generalize on account of given examples and to give accurate answers on data that it has not seen as part of the training process. Any neural network can be made to give accurate answers on the data used for training by making it large enough. Under some circumstances,

neural networks can become over-fitted and thus it will produce good results with the training input data, but perform badly with data it has not seen before. Such a neural network is said to give poor generalization. The careful choice of neural network size and the amount of training applied to the neural network are resulting in good generalization. (Department of Trade and Industry 1994, p. 18)

Performance

The performance of a neural network is its accuracy with other data that it has not been trained. It must not be confused with its speed of operation. A neural network that generalizes well will give good performance. (Department of Trade and Industry 1994, p. 19)

The training process makes a neural computer so different from a conventional computer. The conventional computer has to be explicitly programmed, but the neural computer is trained to solve the problem and to take responsibility for its own internal solutions by setting the right weights. (Department of Trade and Industry 1993, p. 2.5)

The nature of the application and factors like the availability of training data, are the main criteria for the suitability of neural computing for a certain application. (Department of Trade and Industry 1994, p. 14)

4.3.1 Self-Organizing Map / WEBSOM

Some neural networks can learn to cluster data into groups, which seem to have similar statistical properties. These are trained by unsupervised learning schemes. The training set for unsupervised learning comprises only inputs without target outputs. The training algorithm for this type of neural network is used to adjust the input weights so that similar inputs will lead to the same output. (Department of Trade and Industry 1994, p. 20)

Academy professor Teuvo Kohonen has created and developed a data processing method called Self-Organizing Map (SOM) which is based on neural computing and is for the automatic organization and visualization of complex data sets. Over 3000 studies concerning SOM have been published so far and the method has spread throughout the world. It has been applied in many different fields, for instance in automatic speech recognition, image analysis, robotics, modeling of industrial processes, telecommunications applications, medical diagnoses and financial analyses. (Websom-ryhmä 1999)

The Self-Organizing Map uses unsupervised learning and it is therefore applicable to different types of problem. It is useful in applications where is important to analyze a large number of examples and identify groups with similar features. (Browne NCTT 1998)

SOM neural networks are particularly effective in applications where there is a severe imbalance in the number of examples from the different groups that need to be identified. In different fault identification, risk analysis or threat classification applications, where may be many examples of non-fault situations but only few examples of faults, SOM networks could be useful. By training only with examples of the normal situations, it can provide a warning of an unequal situation, where the output does not match well to any of the clusters developed during the training process because of the "incorrect" inputs. (Browne NCTT 1998)

The self-organizing map consists of points, each containing a model describing data items. A data item can consist, for example of information on an individual or a country, or of numerical values describing an individual word or document. The models are arranged so that those close to each other on the map represent mutually similar data items. Each data item is situated at a point where the model best corresponds to that item. (Browne NCTT 1998)

A SOM network architecture is illustrated in the figure 7. There is a map of continuous surface where the edges wrap round to join up at the opposite side. A SOM network can be designed to create a feature map with more than two dimensions, if required by the particular problem. (Browne NCTT 1998)

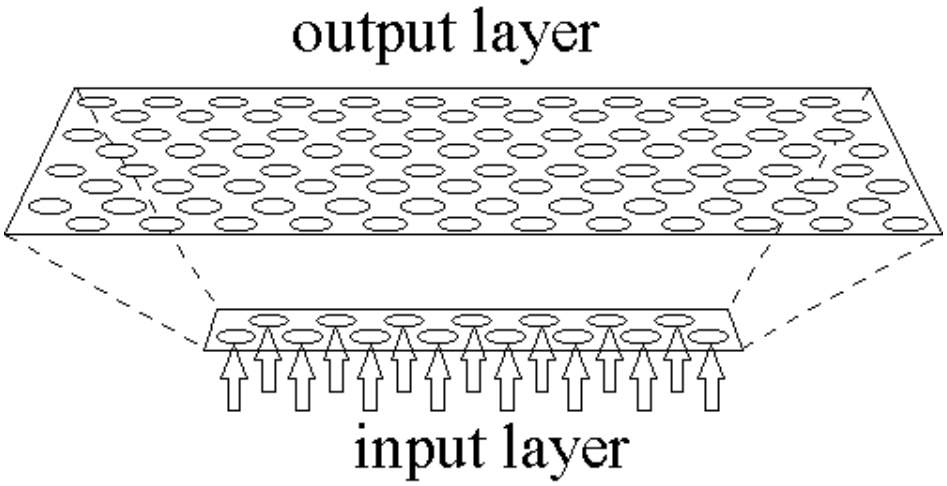

Figure 7. The architecture of SOM network. (Browne NCTT 1998)

WEBSOM is a neural network (SOM) software product for semantic full-text document www-text mining, which means classification, clustering and visualization of multidimensional text data. It automatically organizes documents into 2-dimensional grid so that related documents appear close to each other. It is an application for organizing miscellaneous text documents into meaningful maps for exploration and search. (Tuuli 1998)

4.3.2 Multi-Layer Perceptron Network

The Multi-Layer Perceptron Network (MLP network) has a neural network architecture which comprises a number of identical units organized in layers. In MLP units on one layer is connected to those on the next layer so that the

outputs of one layer are fed-forward as inputs to the next layer. MLP neural networks are typically trained using a supervised training algorithm known as *back propagation*. (Browne NCTT 1998)

A back propagation training algorithm is particularly popular because:

- it is easy to develop neural networks using this method,
- it supports a wide range of applications, especially classification and prediction, and
- most neural network development tools support this algorithm or some of its variations. (Department of Trade and Industry 1993, p. 2.3)

In supervised training, the neural network is presented with a "training set". Each training case comprises an example of the inputs together with the target output. An example of the inputs represents the problem case and the target output represents the known solution. (Department of Trade and Industry 1994, p. 20)

At the time of training, the inputs move over the network and lead to outputs, which are compared with the target output. The weights are adjusted on the neural network's internal connections according to the difference between the input and target sets. By the same token next time a similar training case is presented to the neural network, its actual output will be closer to the target output. The "cost function" measures the difference between a neural network's outputs and the desired outputs. It provides a quantitative measure of the difference between the actual output and the target output. The training continues until the neural network has learnt to produce the required response to within a certain level of tolerance. (Department of Trade and Industry 1994, p. 20)

The "cost function", which is also called the "error function" is an important component to supervised training process. As already mentioned before, it

measures the difference between a neural network's outputs and the desired outputs. It can be applied in three ways:

- During the training process it can be computed for each training case in the training data, and then used to adjust the neural network weights.
- It can be computed for the complete set of training data (by summing the contributions for individual training cases) and used to determine whether training may stop.
- It can be used with the test data as a measure of the acceptability of the results given by the trained network when presented with previously unseen input data. (Department of Trade and Industry 1994, pp. 20-21)

An outline of the back propagation training method is as follows (figure 8.):

1. A set of examples for training the network is gathered. The input into the network consists of a problem statement and the desired output from the network the corresponding solution.

2. The input layer is used for feeding the input data into the network.

3. The input data is processed through the network, layer by layer, until a result is generated by the output layer.

4. The actual output of the network is compared to expected output for that particular input. This results in an *error value,* which represents the discrepancy between given input and expected output. On the basis of this error value all of the connection weights in the network are gradually adjusted, working backwards from the output layer, through the hidden layer, and to the input layer, until the correct output is produced. This is the way to teach the network how to produce the correct output for a particular input, the network *learns*. (Department of Trade and Industry 1993, p. 2.3)

Steps 2 to 4 above are repeated, potentially involving many thousands of examples. As the training process proceeds, the connections, which lead to the correct answers, are strengthened, and the incorrect connections are weakened. (Department of Trade and Industry 1993, p. 2.3)

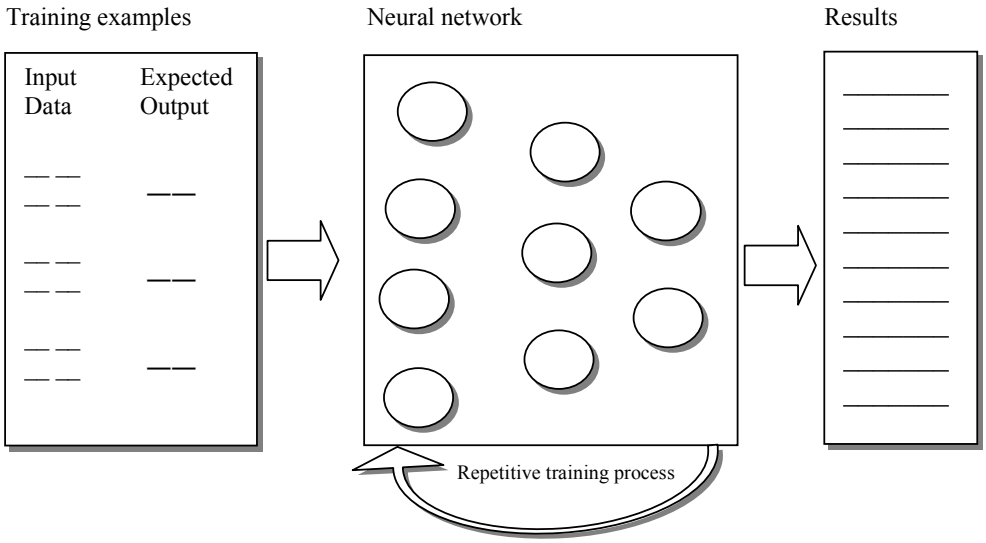

Figure 8. The training Process (Department of Trade and Industry 1993, p. 2.1)

The correct output for every input case produced by the network brings the training process to an end. The network's weights will remain in their trained states and the network is then ready for use. When new input data are presented to the network, the network will determine the output on the basis of its training. The network isn't able to give you an absolutely correct answer but it can be a quite optimum or even the best answer. (Department of Trade and Industry 1993, p. 2.4)

4.3.3 Fuzzy clustering

The creator of the theory of fuzzy systems is professor Lofti Zadeh, who has worked since 1959 in University of California (Berkeley) in United States. In 1960's and 1970's he introduced fuzzy set theory and fuzzy logic basis. (Isomursu et al. 1995, p. 3)

In fuzzy logic natural words are used like for example: *cold, warm, hot, small, average, large*. Those terms are not precise and that's why they cannot be represented in normal set theory, while fuzzy sets allow members to be partial members as well as full members of the set. (Anonymous 2000)

Theory of fuzzy systems is based on fuzzy set theory. When in conventional set, theory objects either belong or do not belong to a set. In fuzzy set theory, an object can also belong partly to some set. This means for instance that a gray object belongs partly in black objects as well as in white objects. (Isomursu et al 1995, p. 8). According to fuzzy set theory an element can belong to different classes at the same time with different grades of membership. (Aminzadeh and Jamshidi 1994, p. 32)

Fuzzy Logic is used in the controlling of subway systems and complex industrial processes, as well as in household and entertainment electronics, diagnosis systems and other expert systems. (Tizhoosh 2000)

Fuzzy Logic is basically a multivalued logic. It allows intermediate values to be defined between evaluations like yes and no, true and false, black and white, etc. According to fuzzy logic it is possible to formulate notions like rather warm or pretty cold mathematically and thus processed by computers. This is the way fuzzy logic makes it possible to program the computers more human-like way. (Tizhoosh 2000)

The very basic notion of fuzzy systems is a fuzzy subset, what we call crisp sets.

Here is an example:
We appoint a set X of all real numbers between 0 and 10. It is the universe of discourse. Then we define a subset A of X of all real numbers between 5 and 8.

X = [0, 10], A = [5,8]

After that we show the set A by its characteristic function. This function assigns a number 1 or 0 to each element in X, depending on whether the element is in the subset A or not. The result can be seen in figure 9. (Tizhoosh 2000)

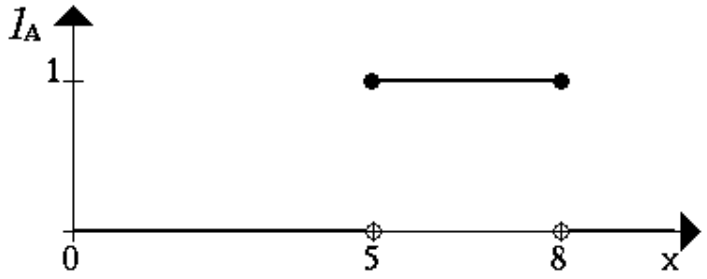

Figure 9. A characteristic function of the set A. (Tizhoosh 2000)

The elements that have assigned with the number 1 include in the set A and the elements that have assigned the number 0 don't include in the set A. Sometimes this concept is sufficient, but often we find that it lacks of flexibility that is necessary in many kinds of applications. (Tizhoosh 2000)

In the next example one kind of problem is shown and the set of young people is described in fuzzy way. We define a subset B to be a set of young people. The lower range of the set is at 0 and the upper range is at 20. Crisp interval of B is between 0 and 20.

B = {set of young people} $\rightarrow$ B = [0,20]

The question is: why is somebody on his 20th birthday young and on the next day not young? To increase the upper bound doesn't give a solution to the question. To broaden the strict limits between young and not young is a natural way to describe whether someone is young or not young. This can be done by using more flexible phrases like: he belongs a little bit more to the set of young people or he belongs nearly not to the set of young people. (Tizhoosh 2000)

Before we coded all the elements of the universe of discourse with 0 or 1. Now we allow all values between 0 and 1. This will make computers process smarter.

I = [0,1]

The interpretation of the numbers is now different than before. The number 1 still means that the element is in the set B and 0 means that the element is not in the set B. All other values mean a gradual membership to the set B. (Figure 10). (Tizhoosh 2000)

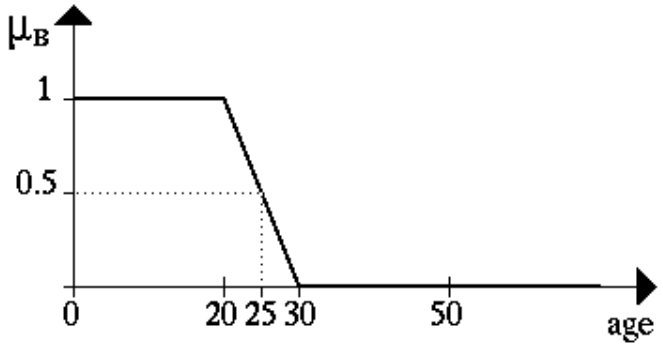

Figure 10. A characterizing membership function of young people's fuzzy set. (Tizhoosh 2000)

Like performed before a 25-year-old one would still be young to a degree of 50 percent. A 30 year old isn't young at all and 20 years old is entirely young. This is the question of a fuzzy set. (Tizhoosh 2000)

5 INFORMATION RETRIEVAL IN IP NETWORKS

The aim of this chapter is to explain, in addition to the main techniques, that search engines use at present in indexing and classification also the deficiencies these techniques have.

There are two major categories of searching tools on the Web: directories, in other words indexes, and search engines. Both require an indexing system done by either a human or a computer. There is a software program called a robot, spider or wanderer, which visits sites and gathers information for computers. (Lager 1996)

The word catalogue is used to imply some form of subject structuring like library catalogue. There are some search engines in the web, which use a catalogue like structure. Because they all use different categories, it is difficult for users to compare gained results. (Wallis et al. 1995)

5.1 Classification at present

The Internet and particularly its collection of multimedia resources known as the WWW, was not designed to support the organized publication and retrieval of information, as libraries are. It has become a disorganized storage for the world's digital publications within reach for everyone. (Lynch 1997)

The use of classification schemes offers one solution for improved access to WWW resources. Web sites have been created to act as a guide to other Web sites. Some of these sites consist of an alphabetical list of subjects. Below each letter is listed the selected Web resources. (Brümmer et al. 1997c)

The nature of electronic indexing can be understood by examining the way Web search engines construct indexes and find information requested by a user.

Periodically, they dispatch programs (crawlers, spiders or indexing robots, as mentioned before) to every site they can identify on the Web. Each site is a set of documents, called pages that can be accessed over the network. The Web crawlers download and examine these pages and extract indexing information that can be used to describe them. (Lynch 1997) Indexing process methods can vary among search engines.

The data gathered by search engines is then stored in the search engine's database, along with an URL, that represents the location where the file resides. After the user submits a search query to the search engine's database, the query produces a list of Web resources, the URLs that can be clicked on to connect to the sites identified by the search. (Lynch 1997)

Almost all automated tools for locating information on the Web are keyword indexes. Classified tools nearly always require some degree of manual input for specifying the appropriate category and other metadata. Many automated search engines have deployed traditional IR indexing strategies and retrieval mechanisms but very few have experimented with automatic classification. (Jenkins et al. 1998)

IR approaches to automatic classification have involved teaching systems to recognize documents belonging to particular classification groups. This have be done by manually classifying a set of documents and then presenting them to the system as examples of documents that belong to each classification. Then the system has built class representatives each of which consists of common terms occurring in the documents known to belong to a particular classification group. When the system subsequently has encountered new documents it has measured the similarity between the document and the class representatives. Each time a new document has been classified it is used to modify the class representative to include its most commonly occurring keywords. (Jenkins et al. 1998)

5.1.1 Search alternatives

Query is the way a user can perform to search engine when trying to find certain information from the web. All major search engines have, besides a simple form of query with one or several keywords, also some additional search options. The scope of these features varies significantly, and there is no standard syntax for invoking them yet established. (Kirsanov 1997b)

Below is a list of some of the most common search options:

- Boolean operators: `AND` (find all), `OR` (find any), `AND NOT` (exclude) to combine keywords in queries,
- phrase search: looking for the keywords only if they are positioned in the document next to each other, in this particular order,
- proximity: looking for the keywords only if they are close enough to each other,
- media search: looking for pages containing Java applets, Shockwave objects, and so on,
- special searches: looking for keywords or URLs within links, image names, document titles,
- various search constraints: limiting the search to a time span of document creation, specifying a document language, and so on. (Kirsanov 1997b)

All search engines organize their results so that at the top of the list will situate more relevant documents. This sorting is based on the frequency of keywords within a document, and the distance of keyword occurrences from the beginning of the document. This means that if one document contains two keywords and another identical document only one keyword, the first document will be closer to the top of list. If two documents are identical except that one has a keyword positioned closer to the top, possibly in the document title, this document will come first. (Kirsanov 1997b)

In addition to these principles, some search engines use extra factors to determine the ranking order. Some of them favor those documents that make use of `META` tags. Some of them rely on link popularity: if a page is linked frequently from other pages and sites, it gets some priority on the list of results. Those that are combinations of a search engine and a directory gives preference to pages reviewed in its own directory. (Kirsanov 1997b)

The lists of search results can be composed of document titles, URLs, summaries, sometimes dates of the document creation and document sizes and several solutions have developed for compiling document summaries. (Kirsanov 1997b)

If pages include meta descriptions provided by page authors, many search engines use them. If metadata is unavailable, search engines usually take the first 100 or 200 characters of page text. There is also search engines which ignores meta tags employing an algorithm that extracts sentences appearing to be the "theme" of the page and presents them as the page's summary. (Kirsanov 1997b)

5.1.2 Searching problems

Existing search engines serve millions of queries a day. Yet it has become clear that they are not ideal for retrieving the amount of continuously growing information on the Web. In contrast to human indexers, automated programs have difficulties to identify characteristics of a document such as its overall theme or its genre like whether it is a poem or a play, or even an advertisement. (Lynch 1997)

Publishers sometimes blame the uncritical characters of automated indexing. A search engine will display first the URLs for the documents that are mentioned a search term most frequently. By repeating the same word within a document, a web site can affect the selection process and attract the search engine’s attention

to the document. In contrast, humans can easily see around simpleminded tricks. (Lynch 1997)

The "information overload" problem is one of the difficulties in searching content on the WWW. Users are frequently overwhelmed by the amount of information available. It is hard for them to filter out the unnecessary information and focus on what is important, and also to actively search for the right information. (Ferguson and Wooldridge 1997)

In addition to user-level problems, there are also a number of organizational factors that make the WWW difficult or inefficient to use. Among other things, there are no widely used standards for metadata or semantic markup, which would allow content providers to annotate their pages with content information. According to Ferguson and Wooldridge (1997), there have been some good reasons for this, chief among them being that beyond obvious metadata elements such as author, title, format, and date, there is no real consensus on what is useful or practicable to associate with a given document. There are also significant technical problems with formalisms for defining document content. W3C organization has tried to solve these problems with XML and RDF.

One problem in information retrieval methods is often their language dependency. For every language a separate term list, document collection and handling is needed. Another problem is the stemming of words, which requires the use of language analysis tools of many languages. These problems can be solved by generating metadata which is suitable and unified for all different media types. If we can create metadata that exactly and descriptively reflects the content of a multimedia document, we can use the created metadata for matching. Also we can create new, more meaningful ways for matching user profiles and multimedia documents if we are able to use structured metadata and a shared ontology to describe the multimedia content. (Savia et al. 1998)

When considering the text classification and its problems, we must take into account also the other content types like images, audio and video. These will be as important part of the content as text and the classification methods of different content types must co-operate well in the future.

According to Savia et al. metadata must be capable enough to support structured data with different types of values to be effective in content describing. Its format must be flexible and expressive to be suitable for different needs like describing different media formats as well as material in several languages. (Savia et al. 1998)

Also a common vocabulary or ontology to describe different types of content is needed. Content providers must produce metadata in compatible metadata formats using shared or compatible ontologies. Otherwise the documents cannot be compared. (Savia et al. 1998)

Katherine Curtis et al. claims that in order to be effective and flexible it must be possible to separate physically media from its metadata, temporarily or permanently. This separation will confuse the quality in other parts of the service if, instead of handling one media file, it is necessary to handle both the media file and an associated file containing its metadata. They have done experiments in order to reduce the size of content items by replacing metadata information with URLs that point to where the information is actually stored. According to them this allows copies of the media to share one physical instance of the metadata information and can provide excellent control for the maintenance and rights management of such information. (Curtis et al. 1999)

One of the biggest problems in use of metadata is that a long time will pass before a reasonable number of people will start to use metadata to provide a better Web classification. The second big problem is that none can guarantee that a majority of the Web objects will ever properly classified via metadata. (Marchiori 1998)

Finally, the WWW was not designed to be used in a methodical way. The process of using the WWW is known as browsing not reading or researching. Browsing is not generally appropriate when attempting to answer complex, important queries. (Ferguson and Wooldridge 1997)

5.2 Demands in future

To realize the potential of the Internet, and overcome the limitations, we require a framework that gives rich, personalized, user-oriented services. To go through with the information overload problem users must be able to find information they really want to find and protect themselves from information they do not want. (Ferguson and Wooldridge 1997)

To find desirable information easily, needs indexing as a minimal service of library. Some libraries like to offer better services. A good librarian would know about the library's users, and use this knowledge in their interests. He can offer other services, not simply search for books or articles when specifically asked. He would pro-actively offer relevant content to a user, as it becomes available. The only that kind of service Internet can offer for the time being are fairly crude search facilities. (Ferguson and Wooldridge 1997)

Mantra Technologies, Inc. has defined that the next generation of web services will include following properties:

- web sites that respond automatically to users' interests with relevant content,
- new ways to surf that help users find the most interesting content on sites,
- easy-to-implement, easy-to-use pinpoint personalization,
- tightly knit communities, built by advanced web services which bring like-minded people together. (Advanced Online Services)

Curtis et al. would use metadata for better the quality of the Internet services and they want to find a way to give to the content the ability to be “self aware”. This will make the use of metadata much larger than just for classification. One illustration of this is adding to a content's metadata throughout its lifetime, whereby the content knows where it has been, in which applications it has been used, how often the content has been copied, who has paid for a copy, and other necessary and beneficial things. This type of information can be used when planning and evaluating new multimedia services. Methods for securing metadata information to prevent the tampering of it and the use of metadata information to control access and copy privileges are important as well. (Curtis et al. 1999)

6 CLASSIFICATION AND INDEXING APPLICATIONS

The purpose in this chapter is to introduce the existing applications about classification. There are some defensive opinions for separate methods. First are introduced applications of library classification, one of each method. All three major library classification schemas are quite widely used and functioning application exists. Next applications introduced are based on neural network methods which are also widely used techniques in classification. The rest of the applications are experiments of miscellaneous systems, which are trying to make accurate information easier to find.

The advantages of document clustering and classification over keyword based indexes have been debated in Information Retrieval (IR) research for quite some time. Documents sharing the same frequently of keywords and concepts are usually relevant to the same queries. Clustering such documents together enables them to be retrieved together more easily and helps to avoid the retrieval of irrelevant unrelated information. Classification usually enables the ability to browse through a hierarchy of logically organized information. This is often considered a more intuitive process than constructing a query string and keyword indexes are comparatively simple to construct automatically. (Jenkins et al. 1998)

6.1 Library classification-based applications

Real world applications of digital libraries are many and varied. They leverage online access to numerous document collections like those own by the worlds academic and public libraries. The digital library technology will play a central role in promoting of other applications such as distance learning, electronic commerce, and intranet knowledge management. (Ferguson and Wooldridge 1997)

Classification schemes like DDC have been responsible for the organization of huge amounts of information for decades. Search engines have inherited from Library Science two main concepts:

1. Metadata - information describing each resource including a description and keyword index terms comparable to a traditional library catalogue entry.
2. Classification - a structured framework for clustering documents that share the same subject. (Jenkins et al. 1998)

6.1.1 WWLib – DDC classification

The notion that Library Science has a lot to offer to the information resource discovery on the Web has evolved to the use of DDC to organize data in WWLib. The original version of WWLib relied on a large degree of manual maintenance and it was a classified directory that was organized according to DDC. Like classified directories in general it offered an option to browse the classification hierarchy or to enter a query string. (Jenkins et al.1998)

The experimental WWLib emphasized the need for automation. Automation has become necessary for any search engine that incorporates a robot or spider, which automatically collects information by retrieving documents and analyzing them for embedded URLs. Other automated components such as an automatic indexer and an automatic classifier were also required. Manually maintained directories have been previously classified and automated search engines have been just huge indexes. Although WWLib utilizes an automated search engine it has decided to maintain the classified nature. (Jenkins et al. 1998)

An outline design of the automated WWLib, shown in figure 11, identifies the automated components and their responsibilities. (Jenkins et al. 1998)

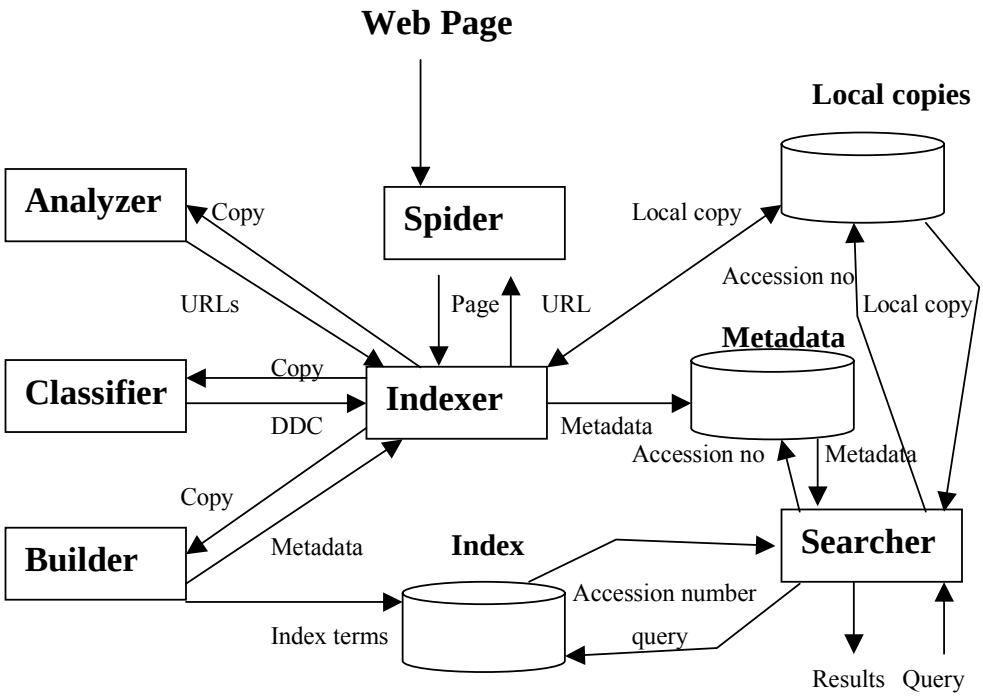

Figure 11. Overview of the WWLib architecture. (Jenkins et al. 1998)

There are six automated components in WWLib system:

1. A **Spider** automatically retrieves documents from the Web.

2. An **Indexer** receives Web pages from the spider. It stores local copies, assigns to them a unique accession number and generates new metadata templates. It also distributes local copies to the Analyzer, Classifier and Builder and adds subsequent metadata generated by the Classifier and the Builder to the assigned metadata template.

3. An **Analyzer** analyses pages, provided by the indexer for embedded hyperlinks to other documents. If found, URLs are passed to the indexer where they are checked that they are pointing to locations in the UK, before being passed to the Spider.

4. A **Classifier** analyses pages provided by the indexer and generates DDC classmarks.

5. A **Builder** analyses pages provided by the indexer and outputs metadata which is stored by the indexer in the document's metadata template. It is also

used to build the index database that will be used to associate keywords with document accession numbers.

6. A **Searcher** accepts query strings from the user and uses them to explore the index database built by the builder. The resulting accession numbers are used to retrieve the appropriate metadata templates and local document copies. Then the searcher uses all this information to generate detailed results, ranked according to relevance to the original query. (Jenkins et al. 1998)

The design of the system is based on an object-oriented design. The classifier object uses a thesaurus with a long list of keywords and synonyms for each classmark. These lists are referred to as class representatives. The classifier would begin the classification process by matching documents against very broad class representatives (containing keywords and synonyms) representing each of the ten DDC classes at the top of the hierarchy listed below:

000 Generalities.
100 Philosophy, paranormal phenomena, psychology.
200 Religion.
300 Social sciences.
400 Language.
500 Natural sciences and mathematics.
600 Technology (Applied sciences).
700 The arts, Fine and decorative arts.
800 Literature (Belles-lettres) and rhetoric.
900 Geography, history, and auxiliary disciplines. (Jenkins et al. 1998)

The classifier takes as its parameter a document object and compares its keywords with the class representatives of the ten top DDC classes. For each DDC class, the matched keyword scores are added to the matched keyword scores from the document. If the resulting value is valid the matching process will then proceed recursively down through the subclasses of those DDC classes that were found to have a valid match score with the document, or a valid

measure of similarity. Documents matched against broad lists of keywords are filtered through sub-classes with more detailed focused terms. (Jenkins et al. 1998)

Jenkins mentioned one problem in this system, which is the low execution speed. According to her the reason for this is the use of Java and the numerous thesaurus information to process. (Jenkins et al. 1998)

The following facilities are available in the WWLib system:

- ***Search*** the catalogue using a **keyword or keywords**.
- ***Search*** for entries by entering a specific **Dewey Decimal Code**.
- ***Read*** the catalogue or sections of the catalogue.
- ***Browse*** the catalogue, using our unique expanding classification browser.
- ***Look*** for WWW pages at a particular site - useful if you know part of a URL. (Burden 1997)

WWlib is run from the School of Computing & Information Technology at the University of Wolverhampton in United Kingdom. It is a searchable, classified catalogue of WWW pages at sites in the United Kingdom. It is organized by using the 20th version of the Dewey Decimal Classification. The Dewey Decimal Classification is copyrighted by Forest Press of Dublin, Ohio, USA, a division of the Online Computer Library Centre of New York, USA and is used with their permission. (Burden 1997)

Other Web resources classified by Dewey:

- Cooperative Online Recourse Catalog (CORC) http://www.oclc.org/oclc/corc/index.htm
- La Trobe University: Bendigo http://library.bendigo.latrobe.edu.au/irs/webcat/ddcindex.htm
- National Library of Canada http://www.nlc-bnc.ca/caninfo/ecaninfo.htm

- Schools' Online Resources for Teachers (SORT) http://www.campus.ort.org/Library/frame.asp?Page=info.asp. (OCLC 2000)

6.1.2 GERHARD with DESIRE II – UDC classification

The UDC-classification is used in the enlarged and multilingual version of the Zürich's Eidgenössische Technische Hochschule (ETH) library in the Deutsche Forchungsgemeinschaft's (DFG) funded project German Harvest Automated Retrieval and Directory (GERHARD). The aim of the project was to establish a service for searching and browsing German Internet resources. Gerhard should automatically collect, classify and index scientifically relevant information in the German WWW. The documents are gathered by a robot, matched to the UDC entries by computer linguistic algorithms and then, create a searchable index and an automatically generated subject tree. (Brümmer et al. 1997d)

Gerhard has following properties according to its home pages:

- it is a quality service specialized in scientifically relevant information
- it is created automatically and, therefore, it is more extensive,
- it is professional,
- it is multilingual,
- it contains 70.000 categories, and
- it is searchable.

In contrast to conventional search engines and directories Gerhard offers the complete integration of searching and browsing. (Müller and Scherp 1998)

In the European Union Telematics for Research project Development of a European Service for Information on Research and Education (DESIRE) have developed architectures, methodologies and a few test-beds for academic Internet users. (Koch 1998)

In cooperation with the German academic Web index GERHARD, run at Oldenburg University, DESIRE project classified the Engineering pages from the DESIRE II test database according to the ETH Zürich three-lingual edition of UDC. This step mainly involves matching the preprocessed full-text records to a dictionary of about 70,000 UDC captions. Part of the GERHARD approach is the matching heuristics from the Institute of Computer Linguistics, Osnabrück University and linguistic software with morphological analyzers for English and German, produced by the Finnish company Lingsof. (Koch and Ardö 2000)

To GERHARD was submitted 101 082 documents receiving 1 151 872 UDC classifications (11 per document). About 2% of the documents (1 912) did not receive any classification. (Koch and Ardö 2000)

All nine UDC classes receive engineering documents. The main class 6 of engineering in UDC receives 30% of the document classifications. The basic sciences for engineering applications, class 5, receives 23%. Almost half of the classifications are spread across the other main classes, 15% in the social sciences (class 3), 14% in information technology etc (class 0) and 8% in geography/history (class 9). The documents are spread across up to 13 levels of depth. Assignments at level 3 are dominating (especially in main class 3 and 9 where a very large part of the documents in these classes are placed), followed by level 4 and 6. The distribution of the classification across the UDC system is shown in appendix 2. (Koch and Ardö 2000)

The largest 15-20 classes were very large including partly more than 5000 member documents. The largest 200 classes include more than 500 documents. The rest of the documents spread across 4000 classes of UDC (among 70 000 classes available). (Koch and Ardö 2000)

In UDC there are many more classes related to engineering available than in Engineering Information Classification Codes (Ei classification). It seems that probably too large a part of the classifications reside in areas of the UDC without obvious connection to engineering or without representing an important

focus of the documents. We assume that the main reason is the limited vocabulary available for every class, just the caption, compared to the about 15 thesaurus terms per class in the Ei system. GERHARD works probably well when treating a heterogeneous database of documents from all subject areas (as in the German GERHARD service). A lot of further improvements and adaptations would be needed. (Koch and Ardö 2000)

According to research project DESIRE, at least five Internet services are currently using UDC:

- BUBL,
- GERHARD,
- NISS Information Gateway,
- OMNI and
- SOSIG. (Brümmer et al. 1997d)

6.1.3 CyberStacks(sm) – LC classification

In 1995 CyberStacks(sm), a demonstration prototype database of selected Internet resources in science, technology and related areas, was formally established on the home page server at Iowa State University. It was meant to be a model for access and use of an increasing number of Internet resources. CyberStacks(sm) was created to identify deficiencies in efforts to organize access to Web resources and the insufficiency in Internet directories and search services. CyberStacks(sm) has adapted the LC classification scheme and a new but at the same time conventional functionality as mechanisms for providing enhanced organization and access to definite resources available over the Web. (McKiernan 1996)

CyberStacks(sm) is a centralized, integrated, and unified collection of WWW and other Internet resources categorized using the LC classification scheme. Resources are organized under one or more relevant LC class numbers and an associated publication format and subject description. LC class number are seen

in appendix 3. The majority of resources incorporated within CyberStacks(sm) collection are monographic or serial works, files, databases or search services. All of those resources are also full-text, hypertext, or hypermedia, and attached to research activities or scientific by their nature. A brief summary is provided for each resource. When necessary, specific instructions on using the resource are also included. (McKiernan 1996)

A LCC call number allows users to browse through virtual library stacks to identify potentially relevant information resources. Resources are categorized first within a broad classification and then within narrower subclasses and after that listed under a specific classification range and associated subject description that best characterize the content and coverage of the resource. (McKiernan 1996)

6.2 Neural network classification-based applications

Neural network-based indexing is quite common. Mostly it is used in pattern recognition but also in text classification as a part of some larger system. Fuzzy systems are used as a part of a neural network classifier in general, and called as neuro-fuzzy classifier. One SOM-based classification system and one neural network classification system is introduced in this chapter, which both are used in text classification and achieved good results.

6.2.1 Basic Units for Retrieval and Clustering of Web Documents - SOM – based classification

The purpose of this approach is to expand the functions of 3-dimensional information organizer for web documents based on SOM to reflect the user's point of view and their purposes. The major aims of this experiment are following:

Classification of web documents by user's point of view.

- We intend to reflect the user's point of view and their purposes to be obtained on the SOM map. We introduce a new mechanism in which relevance feedback operations enable the generation of an overview map, which reflects user needs.

Decision of the Basic Units for Retrieval and Clustering of Web Documents (BUWDs') and their clustering

- It can not be known in advance what kind of answer will be returned from the search engines. That's why it is necessary for users to determine BUWDs according to their purposes. BUDWs are in different ways constructed subgraphs consisting of linked web documents. (Hatano et al. 1999)

There have been previous problems in using SOM in classification, such as:

- to see the relation of the keywords on a SOM map is laborious and takes a lot of time.
- the system cannot reflect user's aim and interest to the overview map. (Hatano et al. 1999)

These problems are intended to be solved by providing the system with the BUWDs which are changeable by an user and feedback operations enables the generation of an overview map which reflects the needs of an user. (Hatano et al. 1999)

We adopt the G.Salton's vector space model for characterization of web documents. A feature vector in a vector space represents the characteristic of each document of the given set of documents. (Figure 12) (Hatano et al. 1999)

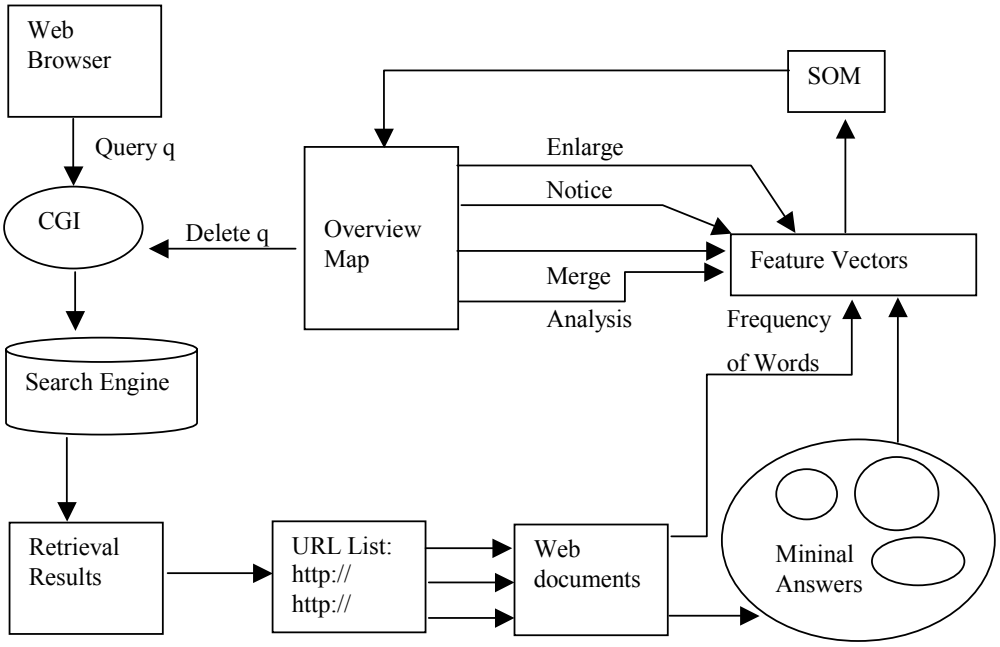

Figure 12. Classification System with BUDWs. (Hatano et. Al 1999)

Web documents obtained from retrieval results of the search engines were used to create an SOM used overview map as follows:

1. The system gets URLs and sources of web documents according to keywords we give to search engine. The results are used like web documents.
2. The words are extracted and ranked from the web documents. Only highly-ranked words are selected.
3. The feature vectors are generated for each web document using two algorithms. One is based on frequency of the words in the web documents, and the other is based on Salton's tf/idf method.
4. Apply the SOM learning algorithm and generate an overview map. (Hatano et al. 1999)

The overview map was done by using Virtual Reality Modeling Language (VRML). It was a 2-dimensional array of cylinders in a hexagonal lattice, where

each cylinder denotes a cell of SOM. Each cylinder has its corresponding cell's feature vector. After applying SOM learning algorithm, each document is mapped to a cylinder. The distance between the cylinder's and the document's feature vector in minimum. If users select the cell in overview map, WWW browser shows URLs and three keywords of web documents, which are classified in the cell. (Hatano et al. 1999)

There are five relevance feedback operations to operate the overview map and to classify effectively web documents:

Enlarge operation

- This operation gathers the web documents, which have something to do with the chosen word and have appeared and clustered in overview map.

Delete operation

- Deletes the web documents in certain keyword area in the overview map, which user judge to be unnecessary. The system and search engine are working together in this operation. This operation is a NOT condition and the user generates a new query q' unconsciously.

Merge operation

- This operation combines some regions. In combining some regions with related words and the system restructures the overview map.

Notice operation

- This operation takes note of the collected web documents. This system restructures an overview map using only web documents in the selected regions.

Analysis operation

- Concerning web documents the language problems might take place because it is possible that the words, which are unfamiliar to user appears on the overview map. This operation could be used to prevent the unfamiliar word clusters of an overview map. (Hatano et al. 1999)

The system consists of three parts: the vector generation part, the SOM learning package called SOM_PAK developed by Kohonen's research group, and the overview map generation part with interactive operations. (Hatano et al. 1999)

Table 1 shows the averages of precision and recall ratios in normal operation and when using relevance feedback operations. Using relevance feedback operations can increase values of precision and recall ratio. This points that the relevance feedback operations make the search of web documents more effective and less laborious. (Hatano et al. 1999)

Table 1. Precision and Recall Ratios between normal and Relevance Feedback Operations (Hatano et al. 1999)

	Precision (%)	Recall (%)
Normal Operation	37.44	34.50
Relevance Feedback Operation	48.82	44.84

This proposed approach uses a sort of neural network technology called SOM to generate a browser for collected web documents with an interactive contents-based similarity retrieval facility. It includes a model for web document's search and retrieval using the minimal answers in making its link structures. (Hatano et al. 1999)

The high effectiveness of this system is showed by the facts:

- The answers of the search engines are provided with the overview map, which make possible to overview the tendency of collected information. Also the labor is reduced in retrieving web documents.

- The system has interaction between the overview map and users. The situation of data clustering is easy to grasp and to improve retrieval efficiency is possible.

- It is possible to retrieve web documents, which we cannot retrieve by search engines. (Hatano et al. 1999)

In spite of the mentioned results there are still problems, which need more researches. Parameters of the feature vectors' generation algorithm should optimized. More experiments to evaluate the quality of the system are needed and to evaluate the system after implementing it by the minimal answers of the web documents. (Hatano et al. 1999)

6.2.2 HyNeT – Neural Network classification

The research project HyNeT has developed a new routing agent for text classification in the Internet. It uses hybrid neural network techniques for classifying news titles as they appear on the Internet newswire. Recurrent plausibility networks are extended with a dynamic short-term memory, which allows the processing of sequences in a robust manner. (Wermter et al. 1999)

The Reuters text categorization test collection, which contains real-world documents, has used for learning the subtask of text routing. The containing documents have appeared on the Reuters newswire and all the titles have belonged to one or more of eight main categories. The categories are Money/Foreign Exchange (money-fx), Shipping (ship), Interest Rates (interest), Economic Indicators (economic), Currency (currency), Corporate (corporate), Commodity (commodity), Energy (energy). (Wermter et al. 1999)

There are abbreviated phrases or sentences, specific characters, and terms and incompleteness or ambiguity in the category assignments. All documents have a title and at least one topic. All 10 733 titles were used. The total number of words was 82 339 and the number of different words in the titles was 11 104. For the training set, 1 040 news titles were used including 130 titles for each of the eight categories. All the rest 9 693 were used for testing. Because one title

can be classified in more than one semantic category, the total number of training titles is 1655 and the total number of test titles is 11 668. The distribution of the titles is seen in table 2. (Wermter et al. 1999)

Table 2. Distribution of the titles. (Wermter et al. 1999)

Category	**Training titles**	**Test titles**
	Number	**Number**
Money-fx	286	427
Ship	139	139
Interest	188	288
Economic	198	1099
Currency	203	92
Corporate	144	6163
Commodity	321	2815
Energy	176	645
All titles	**1040**	**9693**

The used recurrent plausibility network had two hidden and two context layers. Inputs were the word representations and outputs were the desired semantic routing categories. Training is performed until the sum squared error stops to decrease. (Panchev et al. 1999)

In table 3 the obtained recall and precision rates are seen. The generalization performance for new and unseen titles has been even better than the performance on the training data. It demonstrates that overfitting on the training set does not exist. (Panchev et al. 1999)

Table 3. Results of the use of the recurrent plausibility network. (Panchev et al. 1999)

Category	**Training set**		**Test set**	
	Recall	**prec.**	**Recall**	**prec.**
Money-fx	87,34	89,47	86,03	76,70
Ship	84,65	89,21	82,37	90,29
Interest	85,24	87,77	88,19	86,05
Economic	90,24	91,77	81,89	83,80
Currency	88,89	91,36	89,64	89,86
Corporate	92,31	92,66	95,55	95,43
Commodity	92,81	93,14	88,84	90,29
Energy	85,27	87,74	87,69	92,95
All titles	**89,05**	**90,24**	**93,05**	**92,29**

The recall and the precision rates of test set were quite high, 92 % and 93 %. According to Wermter et al. this kind of neural network architecture seems to have a lot of potential for building semantic text routing agents for the Internet in the future. (Panchev et al. 1999)

6.3 Applications with other classification methods

There are numerous amounts of different classification experiments in the field of text classification. Three interesting and new systems are introduced in this chapter. All of them work some how and their functionality can be approved in their web sites.

6.3.1 Mondou – web search engine with mining algorithm

Mondou is a search engine, which apply techniques of text data mining to the web resource discovery. It provides association rules with text data mining techniques and the modification of initial submitted query is possible with boolean expressions including a few keywords. (Kawano and Hasegawa 1998)

Mondou is applied with algorithm for weighted association rules. Since collecting documents are written in Japanese, also morphological analysis and other heuristic operations are used to derive Japanese keywords. Mondou is also called RCAAU, which stands for “retrieval location by weighted association rule” in the digital “monde” and is pronounced “Mo-n-do-u”. (Kawano and Hasegawa 1998)

The web space is regarded as a simple text database without document clusters and doesn’t have any integrated data model. The algorithm that Mondou uses extends and derives association rules. Rules are derived from the huge amount of text data stored in the Internet. This associative mining algorithm is extended again to weighted association rule, which is for handling markup language and especially for HTML tags. (Kawano and Hasegawa 1998)

The agent collects web pages in the Internet and stores them into the text database. It parses collected documents by several methods including natural language processing for Japanese documents and in order to collect more interesting documents the agent often visits to special URLs, if they are referred many times from other web pages. (Kawano and Hasegawa 1998)

The database stores keywords and the number of links from other URLs. Several tables are prepared for different attributes, keywords, URLs hyper links, http servers, stop words, IP addresses and control/management data for Mondou system. (Kawano and Hasegawa 1998)

The query server executes the search program by CGI providing search results and mining association rules to the users. It is possible to enter any suitable combination of search keywords using boolean expressions. After submitted initial keyword, Mondou proposes several associative keywords for user to grasp. (Kawano and Hasegawa 1998)

The Mondou system is described in figure 13 and consists of three main modules: agent, database and query server.

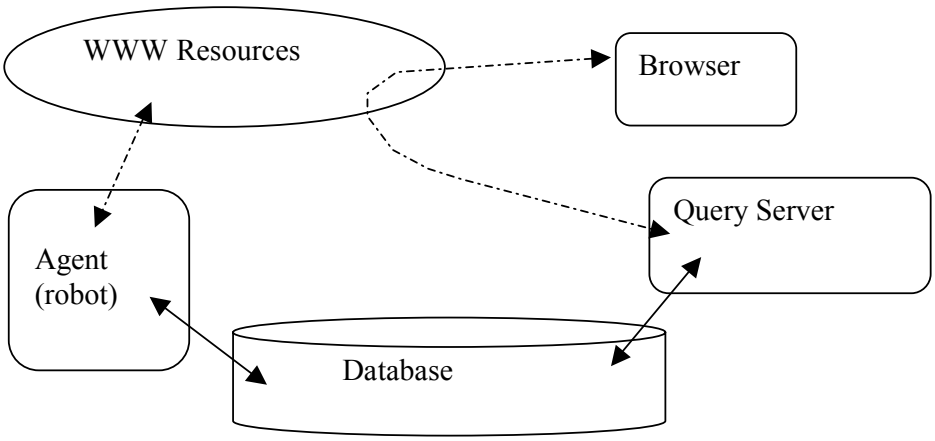

Figure 13. The structure of Mondou system (Kawano and Hasegawa 1998)

Mondou can offer to users a system where they can focus on the URLs appropriately by applying associative rules. The users are not depending conceptual trees or meta knowledge. The expanded algorithm works effectively in searching text data in the web. So far the systems for Mondou have been centralized, but the distributed systems are in plans in order to keep more URLs and to focus on URLs effectively. (Kawano and Hasegawa 1998)

The implementation of Mondou could be seen in site:
http://www.kuamp.kyoto-u.ac.jp/labs/infocom/mondou/index_e.html

6.3.2 EVM – advanced search technology for unfamiliar metadata

The objective of this Defense Advanced Research Projects Agency-funded (DARPA) research project is to link ordinary language queries to unfamiliar indexes and classifications and develop advanced search capabilities to discover and navigate unfamiliar metadata. Using Entry Vocabulary Modules (EVM) associations is created between ordinary language and domain-specific technical metadata vocabulary used to describe databases. (Gey et al. 1999)

Database indexes quite often use individual classification systems and unfamiliar terminology. Because database users are often unfamiliar with these terminologies and classification schemes, they have difficulties in choosing search terms that have actually used in indexing the database. (Chen at al. 1999)

Although there are always wise searchers who would know where to look and how to search, the rest of searchers will benefit from the help in searching. The objective of an Entry Vocabulary Module is to help searchers to convert queries in their own language into the terms used in unfamiliar indexes and classifications (metadata). The purpose of an EVM dictionary is to lead from words or phrases familiar to the searcher to the associated terms in the index or classification to be searched. Also, the feasibility of using intelligent agent software to reduce the amount of human expertise is required when developing these EVM dictionaries is examined. (Lam et al. 1999)

Entry Vocabulary Modules are built to respond adaptively to a searcher's query that is suggested in ordinary language. They are to facilitate a more direct connection between ordinary language queries and unfamiliar indexing terms actually used in organizing information in a variety of unfamiliar databases. This is accomplished by responding to the ordinary language query with a ranked list of terms that may more accurately represent what is searched for than the searcher's choice of terms. These EVMs can serve both as an indexing device and a search aid. (Chen et al. 1999)

The prototypes of Entry Vocabulary Modules are web-accessible at <http://www.sims.berkeley.edu/research/metadata/oasis.html>. They include English language indexes to BIOSIS Concept Codes, to the INSPEC Thesaurus, and to the U.S. Patent and Trademark Office Patent Classification and a multilingual index (supporting queries in English, French, German, Russian, or Spanish) to the physical sciences sections of the Library of Congress Classification. Extending the search to a remote database is restricted to the Patent and Library of Congress classifications because of licensing restrictions, unless the searcher has a Berkeley IP address. Recently has an English language index to the Standard Industrial Classification (SIC) codes been added and it will be linked to numeric database. (Buckland et al. 1999) An entry vocabulary module to map from ordinary language to Foreign Trade Import/Export data classified by the International Harmonized Commodity Classification System (16,000 categories of commodities) is under construction. (Gey et al. 1999)

Associations between ordinary language and domain-specific technical metadata vocabulary are created with the process of Bayesian network. Sufficient training data consisting of document texts is downloaded from the database to provide a probabilistic matching between ordinary language terms and specific metadata classifications which have been used to organize the data. (Gey et al. 1999)

The process of developing the entry vocabulary utilizes both natural language processing modules as well as statistical language techniques for extracting key phrases to map to specialized classifications. Also is examined an application to cross-language retrieval, where metadata classifications in one language is tried to connect to documents in another language, which have been indexed using the original language's metadata. (Gey et al. 1999)

Lexical collocation process of two-stages is used in process. The first stage is creation of an EVM, in other words a dictionary of associations between the lexical items found in the titles, authors, and/or abstracts and the metadata

vocabulary (i.e the category codes, classification numbers, or thesaural terms assigned). This is done by using a likelihood ratio statistic as a measure of association. In the second stage, called deployment, the dictionary is used to predict which of the metadata terms best represent the topic represented by the searcher's terms. (Gey et al. 1999)

After the EVM has led to a promising term in the target metadata vocabulary, a search can then be executed using the newly-found metadata in a remote database. (Buckland et al. 1999)

The EVM-system is implemented in the conventional TCP/IP client-server environment and is running on a UNIX platform. The user accesses the EVM-system from Java/Javascript-enabled or standard Web browsers from URL: http://www.sims.berkeley.edu/research/metadata/, and the Web server initiates a session between the user and the EVM-system via a Common Gateway Interface (CGI). (Norgard et al. 1999)

There are seven kinds of agents in the EVM system: entry vocabulary agents, desktop agents, domain agents, data retrieval agents, cleaner agents, extractor agents and builder agents. Agents are designed to possess a different scope of mission, authority, autonomy, resources, and life expectancy. (Gey et al. 1999)

An agent architecture consisting of those seven interacting agents which "divide and conquer" the entry vocabulary technology tasks are seen in figure 14. These agents range from directory recognition to search agents which locate and identify databases through builder agents which create associations from training databases downloaded by data retrieval agents. Desktop agents help the user to define domains of interest and deploy the association dictionaries created by entry vocabulary agents. (Gey et al. 1999)

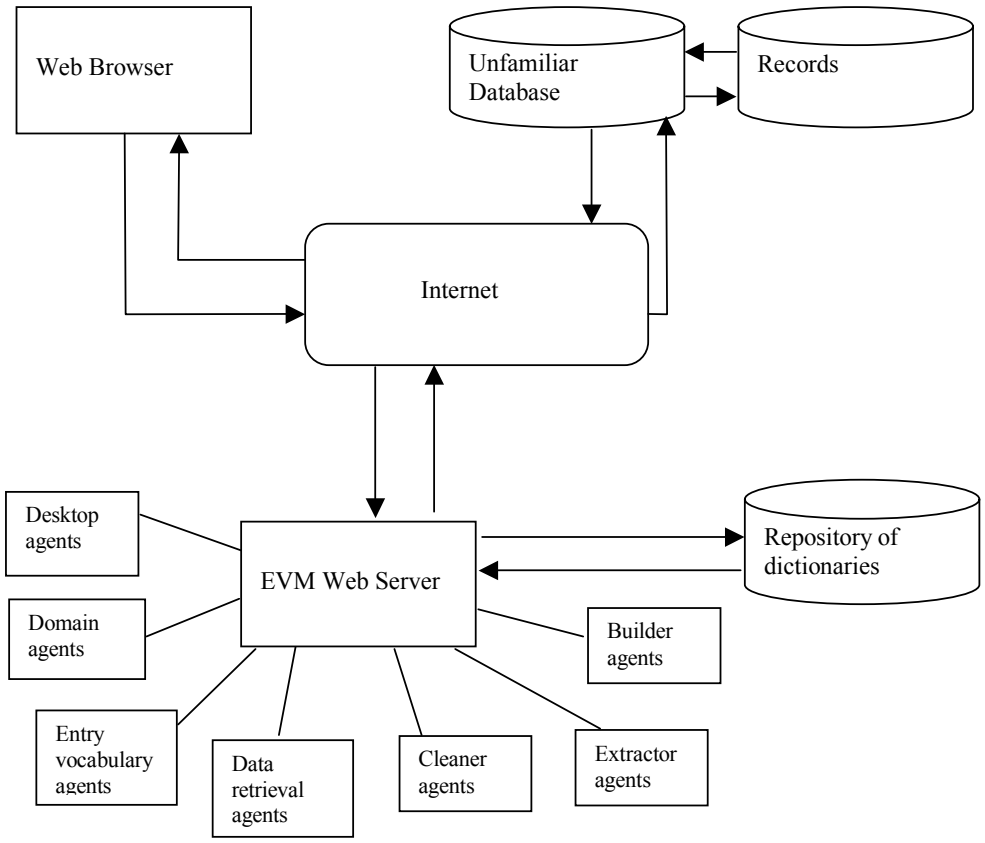

Figure 14. The external architecture of the EVM-system (Gey et al. 1999)

This EVM technique has been applied in addition to textual databases found in the literature, also to numeric databases organized by complex, hierarchical classification schemes. (Gey et al. 1999)

Within the implementation of EVMs some fundamental problems in text categorization relating to the skewedness of the training sets in real-life applications have appeared. New algorithms to achieve greater accuracy in the classification and mapping task are researched. (Gey et al. 1999)

6.3.3 SHOE - Semantic Search with SHOE Search Engine

Simple HTML Ontology Extensions (Shoe) is a search engine, which uses advanced artificial intelligence (AI) technology and tags as is in XML. It allows web page authors to annotate their web documents with machine-readable knowledge. All pages indexed by a search engine must use a special set of tags. These tags describe content and relationships more than just keywords and descriptions. Shoe Search tool allows the user to specify a content for his query and then uses the context to help the user to build a query by example. (Heflin et al. 2000b)

SHOE is an application of SGML and XML. It allows users to define extensible vocabularies and associate machine understandable meaning with them. These vocabularies are called ontologies that consist of as well category as relation definitions. Categories and relation definitions can be augmented with additional axioms as desired. (Heflin et al. 2000a)

In using SHOE, the markup shall be added to the web pages. The user must select an appropriate ontology and then use that ontology's vocabulary to describe the concepts on the page. The process is called annotation. In SHOE each concept is called an instance and is assigned a key. Typically the key is the URL of the web page that best represents the concept and then express that the concept is a member of particular category or that it has certain relationships. (Heflin et al. 2000a)

A tool called Knowledge Annotator is designed for helping authors. It is for adding the knowledge SHOE needs to web pages by making selections and filling in forms. The knowledge SHOE needs is also possible to add with ordinary text editor. The architecture of the SHOE system is illustrated in figure 15. (Heflin et al. 2000a)

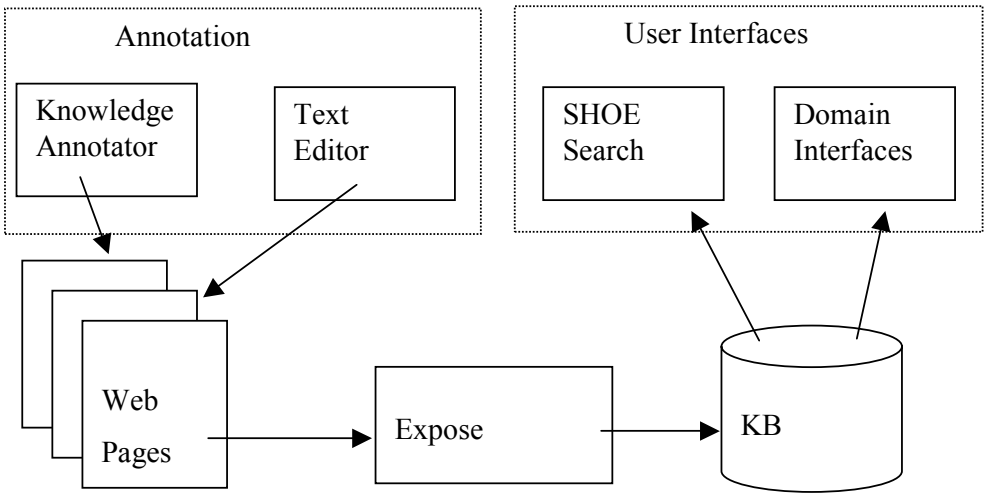

Figure 15. The SHOE system architecture. (Heflin et al. 2000a)

After SHOE pages are annotated and placed on the Web, they can be queried and indexed. A web-crawler called Exposé was developed for collecting the knowledge from the pages and storing it in a repository. It searches for web pages with SHOE markup and interns the knowledge. After finding a new URL Exposé assigns it a cost and uses this to determine where it will be placed in a queue of URLs to be visited. This way the cost function determines the order of the traversal. SHOE pages are assumed to have a tendency to localize and interconnect and this is why, we use a cost function which increases with distance from the start node. It means that paths through non-SHOE pages are more expensive than those paths through SHOE pages. Equally paths that stay within the same directory on the same server are cheaper than those that do not. (Heflin et al. 2000a)

When loading a web page, Exposé parses it. If the ontology the page references to is unfamiliar to Exposé, it loads the ontology as well. It updates its list of pages to visit by identifying all of the hypertext links, category instances, and relation arguments within the page, and evaluates each new URL as above. The agent stores SHOE category, relation claims and new ontology information in a knowledge base (KB). (Heflin et al. 2000a)

In SHOE two ontologies with the same term could mean different things. This is because ontologies can be designed by anyone, and there is no means to prevent terms from being used by multiple ontology authors. Ontologies are renamed by appending an ontology specific string while storing them in the KB. (Heflin et al. 2000a)

When doing a search the user selects a context by choosing an ontology that are known by the underlying KB from a drop-down list. The identifiers and the version numbers of each ontology are displayed, so that the user may choose to issue the query against earlier versions of ontologies. After the user has chosen an ontology, the system creates a list of categories that are defined in that ontology. Subcategories are specializations of categories and indented beneath categories in the list. This taxonomy makes it possible for the user to quickly determine the kinds of objects that are described in the ontology and to choose a convenient class. (Heflin et al. 2000a)

The system responds with a set of properties that are applicable for the category that user has selected. Applicable properties are inheritable. This means that any properties that apply to an ancestor of the selected category are also included in the set. The user can type in values for one or more of the properties, and the system will return only those instances that match all to the specified values. These property values can be literal or instances of classes. (Heflin et al. 2000a)

After the user presses the Query button, the system constructs a conjunctive query and issues it to the KB. The results of the query that KB created are displayed in tabular format. KB can return duplicate results because the same page was visited many times using different URLs. These duplicate results would clutter the display, and therefore they are removed before the system displays them. Both the names and keys are displayed for related instances so that the user can distinguish between instances that happen to have identical names. An instance key the user clicks on is an instance key that matches the

query or one of its properties. The corresponding web page is opened in a new browser window and allows the user to browse the Web with more control over the queries. (Heflin et al. 2000a)

Sometimes users have trouble deciding which values to give for a property and may get no results because of the incorrect values they entered. The Find button has added next to each property to facilitate this problem and find valid values for property in question. By pressing this button, the system will essentially issue a query to find all instances that have a value for the selected property and return those values in the tabular display. The user may then select one of these values and press the Add To Query button to insert it into the query field for the property. The system keeps track of which columns of query results correspond to which properties. (Heflin et al. 2000a)

The user may wish to view the values for a certain property without restricting the query. The Show checkbox allows the user to specify that an additional property should be displayed in the results. This box has checked by default the properties for which the user has specified a value. (Heflin et al. 2000a)

The Show checkbox and the Add To Query button can be used together to help the user gradually filter results and find the desired instances. The user starts by checking some of the Show boxes and issuing a query. One of the results can be selected and added to the query. When the query is reissued, fewer results should be returned and repeating this process the user can continue to reduce the results returned to a bearable set. (Heflin et al. 2000a)

There is also a Web search feature in SHOE that will translate user's query into a similar search engine query and allow them to submit it to any one of the popular search engines. This is for the cases in which all the relevant web pages are not described by SHOE markup. This kind of SHOE search has two advantages over using the search engines directly. At first, the value user entered for the properties increases the chance that the user will provide distinguishing

information for the desired results. Secondly, by automatically creating the query it can take advantage of helpful features that are often overlooked by users such as quoting phrases or using the plus sign to indicate a mandatory term. (Heflin et al. 2000a)

7 CONCLUSIONS

Information retrieval has become very important subject in the evolution of the Internet. It is visible that the service level that search engines are able to offer at present is inadequate for finding relevant, reliable and particular information on the Web. If web shops and content providers are slowly finding out that the money they have invested to their sites does not yield profits, they might cease to provide them. At least the development will slow down. Also for an ordinary user who has not gotten familiar with search engines the searching process is weary, frustrating and prolonged.

Generally there are two kinds of searchers. The first one likes to find something about a specific item, which of he already knows something about. The other one has a particular problem in mind, and wants to find documents, which deal with that problem.

The extent and speed have created a demanding task for automated ways to effectively manage the content on the Web. The Internet will contain diverse types of content like images and videos in addition to text, which must take into account when deciding the classification method of web resources. Also the support for multiple languages should be remembered and the content must be viewable across a variety of browsing devices. Automated indexing techniques are coming absolutely relevant in efficient indexing and classification of Internet resources. The developing algorithms for automatic indexing and classification have a great weight in efficiency of information retrieval.

Library science has a lot to offer to the complicated task of information resource discovery on the Web. Librarians and classification experts in libraries have done a great deal of work in developing the library classification schemes for Internet and especially for virtual library usage. It might be a considerable alternative to combine the skills of the librarians and the computer scientists in developing the solutions for the classification problem in the Internet.

There are many strengths in library classification schemes. They are agreed as international standards which means that they are widely recognized, used and available. They are regularly maintained and not language dependant and already exists in several different languages. However the virtues classification systems have some weaknesses as well. They are too wide and complex to use and some subjects might be out of date or not updated frequently enough.

The use of any hierarchical classification system in Internet service would be useful and offering many advantages. It brings together collections of similar resources and supports the browsing of these collections. The precision of searches is high and many databases can be searched at same time. The favor of Yahoo shows that people like to use hierarchical classification systems.

Because the information retrieval is a popular subject of research at present there are many projects and researches in progress and many of the systems and projects introduced in this thesis are still unfinished. Though library classification systems have many virtues they are not suitable for entire Web. The library classification systems are well in virtual libraries and perhaps could be used in combining individual libraries and thus make possible to search information in same time from several libraries. Otherwise a component architecture that separates content from the presentation format is needed.

Separating content and presentation ensures that site will look right in spite of the device it is looked at. XML, XHTML, metadata and RDF are presumably the key methods in organizing the Internet resources in the future. The use of the metadata with these new methods might resolve also the problem of semantic. In order to become popular these new methods should be developed easy enough to use.

Instead of indexing entire Web in one huge integrated index, a collection of Web indexes and virtual library systems could be a solution. Documents could be

divided in different information spaces, like industry, science, trade and so on and having subclasses under them. Neural network methods and fuzzy systems have used successfully in automatic classification process. They are noteworthy when deciding the custom of automatic classification. The recall and precision property seems to be very useful in search alternatives.

The main conclusion of this thesis is that there is not one good classification method beyond others, but many advanced methods. Combining them some way will be the best possible system in classifying and indexing diverse Internet resources.

In IPMAN project my task was to analyze different classification methods and select one for the needs of the project. Because the basis and the goal of the project changed the selection of the classification method became unnecessary. The new objective of IPMAN is to implement Content Management System performing Content Management. This Content Management System is designed to be integrated with TriMedia service platform and forming together Content Management Framework.

TriMedia is developed by the Student Union of the Helsinki University of Technology and it is a flexible service platform which is independent of the browsing devices. In TriMedia the documents are stored in databases in XML-format. The RDF metada will be generated from these XML-format documents and stored for the possible use of classification system or other needs appearing in the future.

8 SUMMARY

The Internet has become very important source of information for commercial enterprises in addition to being an infrastructure for electronic mail. The increase and diversity of WWW have created an increasing demand for better and versatile information and knowledge management services.

Resource discovery and filtering are becoming more important all the time. Indexing and classification is an important part of resource discovery and filtering of new and changing Internet resources. Though the number of professional and commercially valuable information has increased considerably, the searching and retrieving still relies on general-purpose Internet search engines. New systems are needed to satisfy the varied requirements of users.

The scope of this thesis was to focus on basic classification and indexing methods in IP networks and to some of the latest applications and projects where those methods were used. The main purpose was to find out what kind of applications for classification and indexing have been generated lately and the advantages and weaknesses of them.

Many study groups are trying to develop an efficient and useful classification system to be used in information retrieval in Internet. Librarians and classification experts have done much in developing the library classification schemes for Internet. The use of hierarchical classification system in Internet service would be useful and offering many advantages. Presumably hierarchical classification systems will stay in use of virtual libraries and XML, XHTML, metadata and RDF will be the key methods in organizing the Internet resources and resolving the problem of semantic at the same time. The main conclusion of this thesis was that there is not one good classification method beyond others, but many advanced methods. Combining them some way will be the best possible system in classifying and indexing diverse Internet resources.

REFERENCES

Advanced Online Services. http://www.mantratech.com/whatwedo/11.html Mantra Technologies, Inc. CALIFORNIA

Aminzadeh, F. and Jamshidi, M. 1994. Fuzzy Logic, Neural Networks and Distributed Artificial Intelligence. New Jersey. 301 p.

Anonymous. 2000. Fuzzy / Neurofuzzy Logic. [WWW-document]. Available: http://www.ncs.co.uk/nn_fzy.htm (Cited 26.5.00).

Berners-Lee, T. 1997. Meta Architecture. [WWW-document]. Available: http://www.w3.org/DesignIssues/Metadata (Cited 30.5.2000).

Bos, B. 2000. XML in 10 points. [WWW-document]. Available: http://www.w3.org/XML/1999/XML-in-10-points (Cited 29.5.2000).

Bray, T., Paoli, J., Sperberg-McQueen, C. 1998. Extensible Markup Language (XML) 1.0. W3C recommendation 10-February-1998. [WWW-document]. Available:http://www.w3.org/TR/REC-xml (Cited 29.5.00).

Browne Ray (coordinator). 1998. Neural Computing Technology Transfer. Overview Of Neural Computing. [WWW-document]. Available: http://www.brainstorm.co.uk/NCTT/tech/tby.htm (Cited 25.5.00).

Bryan, M. 1998. An Introduction to the Extensible Markup Languahe (XML). [WWW-document]. Available: http://www.personal.u-net.com/~sgml/xmlintro.htm (Cited 20.6.2000).

Brümmer A., Day M., Hiom D., Koch T., Peereboom M., Poulter A. and Worsfold E. 1997a. The role of classification schemes in Internet resource description and discovery. [WWW-document]. Available:

http://www.lub.lu.se/desire/radar/reports/D3.2.3/f_1.html (Cited 9.3.2000).

Brümmer, A., Day, M., Hiom, D., Koch, T., Peereboom, M., Poulter, A. and Worsfold, E. 1997b. The role of classification schemes in Internet resource description and discovery**.** Current use of classification schemes in existing search services. [WWW-document]. Available: http://www.ukoln.ac.uk/metadata/desire/classification/class_2.htm (Cited 9.2.00).

Brümmer, A., Day, M., Hiom, D., Koch, T., Peereboom, M., Poulter, A. and Worsfold, E. 1997c. The role of classification schemes in Internet resource description and discovery. Introduction and overview. [WWW-document]. Available: http://www.ukoln.ac.uk/metadata/desire/classification/class_1.htm (Cited 29.2.00).

Brümmer, A., Day, M., Hiom, D., Koch, T., Peereboom, M., Poulter, A. and Worsfold, E. 1997d. The role of classification schemes in Internet resource description and discovery**.** Current use of classification schemes in existing search services. [WWW-document]. Available: http://www.ukoln.ac.uk/metadata/desire/classification/class_3.htm (Cited 29.2.00).

Buckland, M., Chen, A., Chen, H., Kim, Y., Lam, B., Larson, R., Norgard, B., and Purat, J. 1999. Mapping Entry Vocabulary to Unfamiliar Metadata vocabularies. D-Lib Magazine, January 1999, volume 5, number 1. [WWW-document]. Available: http://www.dlib.org/dlib/january99/buckland/01buckland.html (Cited 11.9.00).

Burden, Peter. 1997. The UK Web Library - Searchable Classified Catalogue of UK Web sites. [WWW-document]. Available: http://www.scit.wlv.ac.uk/wwlib/ (Cited: 22.8.2000).

Chakrabarti, S., Dom, B., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J. M. and Gibson, D. 1999a. Hypersearching the Web. Scientific American, number 6. [WWW-document]. Available: http://www.sciam.com/1999/0699issue/0699raghavan.html (Cited 9.3.2000).

Chakrabarti, S., Dom, B., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Kleinberg, J. M. and Gibson, D. 1999b. Mining the Web's Link Structure. IEEE, pp. 60-67.

Chen, H-M., Gey, F., Norgard, B., Buckland, M., Kim, Y., Chen, A., Lam, B., Purat, J., and Larson, R. 1999. Search Support for Unfamiliar Metadata Vocabularies. [WWW-document]. Available: http://www.sims.berkeley.edu/research/metadata/index.html (Cited 11.9.00).

Chen, M., Han J. and Yu, P. 1996. Data Mining: An Overview from a Database Perspective. IEEE Transactions on knowledge and data Engineering, volume 8, number 6.

Connolly, D. 2000. XML Pointer, XML Base and XML Linking. [WWW-document]. Available: http://www.w3.org/XML/Linking.html (Cited 21.6.2000).

Curtis, K., Foster, P. W. and Stentiford, F. 1999. Metadata - The Key to Content Management Services. [WWW-document]. Available: http://computer.org/proceedings/meta/1999/papers/56/curtis.html (Cited 14.8.2000).

DCMI. 2000a. Dublin Core Element Set, Version 1.0: Reference Description. OCLC Inc. [WWW-document]. Available: http://purl.org/dc/documents/rec-dces-199809.htm (Cited 1.6.2000).

DCMI. 2000b. The Dublin Core: A Simple Content Description Model for Electronic Resources. OCLC Inc. [WWW-document]. http://purl.oclc.org/dc (Cited 1.6.2000).

DCMI. 2000c. The Dublin Core: A Simple Content Description Model for Electronic Resources. OCLC Inc. [WWW-document]. Available: http://mirrored.ukoln.ac.uk/dc/. (Cited 7.3.2000).

Department of Trade and Industry. 1993. Directory of Neural Computing Suppliers, Products and Sources of Information. London. 80 p.

Department of Trade and Industry. 1994. Best Practice Guidelines for Developing Neural Computing Applications. London. 178 p.

Ferguson, I. and Wooldridge, M. 1997. Paying Their Way. Commercial Digital Libraries for the 21st Century. [WWW-document]. Available: http://webdoc.gwdg.de/edoc/aw/d-lib/dlib/june97/zuno/06ferguson.html (Cited 11.7.2000).

Gardner Tracy. 1999. eLib Web Pages Metadata. [WWW-document]. Available: http://www.ukoln.ac.uk/services/elib/metadata.html (Cited 10.3.2000).

Gey, F., Chen, H-M., Norgard, B., Buckland, M., Kim, Y., Chen, A., Lam, B., Purat, J. and Larson, R. 1999. Advanced Search Technologies for Unfamiliar Metadata. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999. USA, University of California, Berkeley [WWW-document]. Available: http://metaphor.sims.berkeley.edu/papers/gey/advanced-search-ieee.pdf (Cited 11.9.00).

Gudivara, V., Raghavan, V., Grosky, W. and Kasanagottu, R. 1997. Information Retrieval on the World-Wide Web. [WWW-document]. Available: http://www.cacs.usl.edu/~raghavan/guivada-rev.html (Cited 13.3.00).

Hatano,K., Sano, R., Duan,Y. and Tanaka, K. 1999. An interactive classification of Web documents by self-organizing maps and search engines. Proceedings of the 6th International Conference on Database Systems for Advanced Applications. Kobe, Japan. Institute of Electrical and Electronics Engineers, Inc. Pages 35-42.

Heery Rachel. 1998. What is… RDF? [WWW-document]. Available: http://www.ariadne.ac.uk/issue14/what-is/ (Cited 10.3.00).

Heflin, Jeff and Hendler, James. 2000a. Searching the Web with SHOE. [WWW-document]. Available: http://www.cs.umd.edu/projects/plus/SHOE/pubs/aiweb2000.pdf (Cited: 24.8.2000.)

Heflin, Jeff and Hendler, James. 2000b. Semantic Interoperability on the Web. [WWW-document]. Available: http://www.cs.uml.edu/projects/plus/SHOE/pubs/extreme2000.pdf (Cited: 24.8.2000.)

Iannella, R. 1999. An Idiot's Guide to the Resource Description Framework. [WWW-document]. Available: http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/. (Cited 19.06.2000).

IFLA. 2000. DIGITAL LIBRARIES: Metadata Resources. [WWW-document]. Available: http://www.ifla.org/II/metadata.htm (Cited 30.6.2000).

Isomursu, P., Niskanen, V., Carlsson, C. and Eklund, P. 1995. Sumean logiikan mahdollisuudet. 5th edition. Helsinki, TEKES. 100 p.

Jenkins, C., Jackson, M., Burden, P. and Wallis, J. 1998. Automatic Classification of Web Resources using Java and Dewey Decimal Classifications. Proceedings of Seventh International World Wide Web Conference, Brisbane, Queensland, Australia also in Computer Networks and ISDN Systems, Vol. 30, 1998, pages: 646-648.

Kawano, Hiroyuki and Hasegawa, Toshiharu. 1998. Mondou: Interface with Text Data Mining for Web Search Engine. Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98). IEEE Computer Society.

Kirsanov, D. 1997a. HTML Unleashed PRE: Strategies for Indexing and Search Engines. The Meta Controversy. [WWW-document]. Available: http://www.webreference.com/dlab/books/html-pre/43-2-4.html (Cited 12.7.2000).

Kirsanov, D. 1997b. HTML Unleashed PRE: Strategies for Indexing and Search Engines. Search Interface. [WWW-document]. Available: http://www.webreference.com/dlab/books/html-pre/43-1-2.html (Cited 12.7.2000).

Koch Traugott and Ardö Anders. 2000. Automatic classification of full-text HTML-documents from one specific subject area DESIRE II D3.6a. Working Paper 2. [WWW-document]. Available: http://www.lub.lu.se/desire/DESIRE36a-WP2.html#univ (Cited 31.8.2000).

Koch, Traugott. 1998. Desire2-pres.txt. [WWW-document]. Available: http://www.lub.lu.se/tk/desire2/desire2-pres.txt Cited: 22.8.2000).

Kowalski, Gerald. 1997. Information Retrieval Systems. Theory and Implementation. Third Printing. Massachusetts, USA. Kluwer Academic Publishers. 282 pages.

Lager, Mark. 1996. Spinning a Web Search. [WWW-document]. Available: http://www.library.ucsb.edu/untangle/lager.html (Cited 14.8.2000).

Lam, B., Buckland, M., Chen, H-M., Gey, F., Norgard, B., Kim, Y., Chen, A., Purat, J. and Larson, R. 1999. Intelligent EVM Agent. [WWW-document]. Available: http://sims.berkeley.edu/research/metadata/Intelligentagents.htm (Cited 22.9.00).

Lassila Ora and Swick Ralph. 1999. Resource Description Framework (RDF) Model and Syntax Specification. [WWW-document]. Available: http://www.w3.org/TR/PR-rdf-syntax/ (Cited 04.10.2000).

Library of Congress. 1999. Library of Congress Classification Outline. [WWW-document]. Available: http://lcweb.loc.gov/catdir/cpso/lcco/lcco.html and http://lcweb.loc.gov/catdir/cpso/lcco/lcco_t.html (Cited 11.8.2000).

Lilley, C. and Quint, V. 2000. Extensible Stylesheet Language (XSL). [WWW-document]. Available: http://www.w3.org/Style/XSL/ (Cited 21.6.2000).

Lynch Clifford. 1997. Searching the Internet. Scientific American Article.). [WWW-document]. Available: http://www.sciam.com/0397issue/0397lynch.html (Cited 26.9.2000).

Massimo Marchiori. 1998. The Limits of Web Metadata, and Beyond. Available: http://www7.scu.edu.au/programme/fullpapers/1896/com1896.htm Proceedings of the Seventh International World Wide Web Conference (WWW7). Brisbane, Australia. Also in Journal of Computer Networks and ISDN Systems, n.30, pp. 1-9, Elsevier.

McIlwaine, Ia. 1998. Communications and Classification in the Next Century. [WWW-document]. Available: http://www.oclc.org/oclc/research/publications/review98/mcilwaine/commclass.htm (Cited 14.3.00).

McKiernan, Gerry. 1996. Casting the Net: The Development of a Resource Collection for an Internet Database. [WWW-document]. Available: http://www.library.ucsb.edu/untangle/mckiernan.html (Cited: 27.9.2000).

McKiernan, Gerry. 1998. Welcome to CyberStacks(sm)! [WWW-document]. Available: http://www.public.iastate.edu/~CYBERSTACKS/homepage.html (Cited: 27.9.2000).

Müller Gerhard and Scherp Ansgar. 1998. GERHARD - German Harvest Automated Retrieval and Directory. [WWW-document]. Available: http://www.gerhard.de/gerold/owa/gerhard.create_index_html?form_language=1&form_timestamp=&form_language=1 (Cited 24.8.00).

Norgard, B., Lam, B., Buckland, M., Chen, H-M., Gey, F., Kim, Y., Chen, A., Purat, J. and Larson, R. 1999. Entry Vocabulary Modules System Online Help. [WWW-document]. Available: http://sims.berkeley.edu/research/metadata/help.html (Cited 11.9.00).

OCLC Forest Press. 2000. Dewey Worldwide : Web Resources. http://www.oclc.org/fp/worldwide/web_resources.htm (Cited: 22.8.2000).

Panchev, C., Arevian, G. and Wermter, S. 1999. Recurrent Neural Network Learning for Text Routing. Artificial Neural Networks. Conference Publication No. 470. IEEE. P. 898-903.

Patterson, Kate. 1997. Cache Object Content Analysis. [WWW-document]. Available: http://mans.cee.hw.ac.uk/Vortices/discussion/ContentAnalysis.html (Cited 23.3.00).

Plambeck, James. 1996. Universal Decimal Classification Index. [WWW-document]. Available: http://www.chem.ualberta.ca/~plambeck/udc/index.htm (Cited 11.8.00).

Raggett, D. (editor). 1999. Introduction to HTML 4. [WWW-document]. Available: http://www.w3.org/TR/REC-html40/intro/intro.html (Cited 21.3.00).

Raggett, D., Jacobs, I., Ishikawa, M. and Asada, T. 2000. HyperText Markup Language. Home page. [WWW-document]. Available: http://www.w3.org/MarkUp/ (Cited 26.5.00).

Savia Eerika, Kurki Teppo and Jokela Sami. 1998. Metadata Based Matching of Documents and User Profiles. [WWW-document]. Available: http://smartpush.cs.hut.fi/pubdocs/step98/index.htm (Cited 14.8.2000).

Schwartz, Kathryn L. 1997. Reseach & Writing. [WWW-document]. Available: http://www.ipl.org/teen/aplus/dewey1.htm (Cited 11.8.2000).

Stenvall, J. and Hakala, J. 1998. Dublin Core – formaatin käyttöopas. [WWW-document]. Available: http://hul.helsinki.fi/meta/dc-opas.html (Cited 1.6.2000).

Tizhoosh, Hamid R. 2000. Fuzzy Image Process. Introduction of Fuzzy Systems. [WWW-document]. Available: http://ipe.et.uni-magdeburg.de/~hamid/Tutorial2.htm (Cited 26.5.00).

Tuuli, Raimo. 1998. Oppivien ja älykkäiden järjestelmien sovellukset. Vuosiseminaari, Espoo, 22.04.1998. Tekes.

UKOLN Metadata Group. 1997. 2. Current use of classification schemes in existing search services. [WWW-document]. Available: http://www.ukoln.ac.uk/metadata/desire/classification/class_4.htm (Cited 14.3.00).

Uosukainen, L., Lilja, T., Metso, L., Ylänen, S., Ihalainen, S., Karvo, J. and Taipale, O. 1999. IP Network Management. Iterim report of the IPMAN-project. Espoo, Helsinki University of Technology. 92 p.

W3C HTML working group. 2000. XHTML ™ 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0. [WWW-document]. Available: http://www.w3.org/TR/xhtml1/ (Cited 15.6.2000).

W3C HTML working group. 1999. On SGML and HTML. [WWW-document]. Available: http://www.w3.org/TR/REC-html40/intro/sgmltut.html (Cited 17.10.2000).

Wallis, J and Burden, J. 1995. Towards a Classification-based Approach to Resource Discovery on the Web. 4th International W4G Workshop on Design and Electronic Publishing, Abingdon, UK.

Walsh, Norman. 1998. A Technical Introduction to XML. [WWW-document]. Available: http://nwalsh.com/docs/articles/xml/ (Cited 20.6.2000).

Websom-ryhmä. 1999. WEBSOM - itseorganisoivia karttoja dokumenttikokoelmista. [WWW-document]. Available: http://websom.hut.fi/websom/stt/doc/fin/taustaa.html (Cited 20.3.2000).

Weibel, S., Kunze, J., Lagoze, C., Wolf, M. 1998. Dublin Core Metadata for Resource Discovery. [WWW-document]. Available: http://www.ietf.org/rfc/rfc2413.txt (Cited 1.6.2000).

Wermter, S., Panchev, C., and Arevian, G. 1999. Hybrid Neural Plausibility Networks for News Agents. Proceedings of the National Conference on Artificial Intelligence. Orlando, USA. P. 93-98.

Äyväri, H. 1997. Where is the Internet evolving in the near future? [WWW-document]. Available: http://www.tcm.hut.fi/Opinnot/Tik-110.551/1997/internet.htm (Cited 12.7.2000).

APPENDIXES

Appendix 1.

Dewey Decimal Classification System (Schwartz 1997)

000 Generalities	**100 Philosophy and Psychology**
010 Bibliography 020 Library & information sciences 030 General encyclopedic works 040 Unassigned 050 General serials & their indexes 060 General organizations & museology 070 News media, journalism, publishing 080 General collections 090 Manuscripts & rare books	110 Metaphysics 120 Epistemology, causation, humankind 130 Paranormal phenomena 140 Specific philosophical schools 150 Psychology 160 Logic 170 Ethics (moral philosophy) 180 Ancient, medieval, Oriental philosophy 190 Modern Western philosophy
200 Religion	**300 Social sciences**
210 Natural theology 220 Bible 230 Christian theology 240 Christian moral & devotional theology 250 Christian orders & local church 260 Christian social theology 270 Christian church history 280 Christian denominations & sects 290 Other & comparative religions	300 Sociology and anthropology 310 General statistics 320 Political science 330 Economics 340 Law 350 Public administration 360 Social services; associations 370 Education 380 Commerce, communications, transport 390 Customs, etiquette, folklore
400 Language	**500 Natural sciences & mathematics**
410 Linguistics 420 English & Old English 430 Germanic languages German 440 Romance languages French 450 Italian, Romanian languages 460 Spanish & Portuguese languages 470 ltalic languages, Latin 480 Hellenic languages, Classical Greek 490 Other languages	510 Mathematics 520 Astronomy & allied sciences 530 Physics 540 Chemistry & allied sciences 550 Earth sciences 560 Paleontology, paleozoology 570 Life sciences 580 Botanical sciences 590 Zoological sciences

600 Technology (Applied sciences)	**700 The Arts**
600 General technology 610 Medical sciences and medicine 620 Engineering & allied operations 630 Agriculture 640 Home economics & family living 650 Management & auxiliary services 660 Chemical engineering 670 Manufacturing 680 Manufacture for specific uses 690 Buildings	710 Civic & landscape art 720 Architecture 730 Plastic arts, sculpture 740 Drawing & decorative arts 750 Painting & paintings (museums) 760 Graphic arts, printmaking & prints, postage stamps 770 Photography & photographs 780 Music 790 Recreational & performing arts
800 Literature & rhetoric	**900 Geography & history**
810 American literature 820 English & Old English literatures 830 Literatures of Germanic languages 840 Literatures of Romance languages 850 Italian, Romanian literatures 860 Spanish & Portuguese literatures 870 Italic literatures, Latin 880 Hellenic literatures, Classical Greek 890 Literatures of other languages	900 World History 910 Geography and travel 920 Biography, genealogy, insignia 930 History of the ancient world 940 General history of Europe 950 General history of Asia, Far East 960 General history of Africa 970 General history of North America 980 General history of South America 990 General history of other areas

Appendix 2.

UNIVERSAL DECIMAL CLASSIFICATION (Plambeck, 1996)

The Universal Decimal Classification Index

0 Generalities. Information. Organization.

- 1 Philosophy. Psychology.
- 2 Religion. Theology.
- 3 Social Sciences. Economics. Law. Government. Education.
- 4 vacant
- 5 Mathematics and Natural Sciences.
- 6 Applied Sciences. Technology. Medicine.
- 7 The Arts. Recreation. Entertainment. Sport.
- 8 Language. Linguistics. Literature.
- 9 Geography. Biography. History.

Appendix 3.

LIBRARY OF CONGRESS CLASSIFICATION OUTLINE

Listed below are the letters and titles of the main classes of the Library of Congress Classification. Click on any class to view a breakdown of its subclasses. (Library of Congress. 1999)

- A -- GENERAL WORKS
- B -- PHILOSOPHY. PSYCHOLOGY. RELIGION
- C -- AUXILIARY SCIENCES OF HISTORY
- D -- HISTORY: GENERAL AND OLD WORLD
- E -- HISTORY: AMERICA
- F -- HISTORY: AMERICA
- G -- GEOGRAPHY. ANTHROPOLOGY. RECREATION
- H -- SOCIAL SCIENCES
- J -- POLITICAL SCIENCE
- K -- LAW
- L -- EDUCATION
- M -- MUSIC AND BOOKS ON MUSIC
- N -- FINE ARTS
- P -- LANGUAGE AND LITERATURE
- Q -- SCIENCE
- R -- MEDICINE
- S -- AGRICULTURE
- T -- TECHNOLOGY
- U -- MILITARY SCIENCE
- V -- NAVAL SCIENCE
- Z -- LIBRARY SCIENCE

Class T

TECHNOLOGY

```
T
1-995            Technology (General)
10.5-11.9          Communication of technical information
11.95-12.5         Industrial directories
55-55.3            Industrial safety.  Industrial accident
                     prevention
55.4-60.8          Industrial engineering.  Management
                       engineering
57-57.97             Applied mathematics.  Quantitative methods
57.6-57.97             Operations research.  Systems analysis
58.4                 Managerial control systems
58.5-58.64           Information technology
58.6-58.62           Management information systems
58.7-58.8            Production capacity.  Manufacturing
capacity
59-59.2              Standardization
59.5                 Automation
59.7-59.77           Human engineering in industry.  Man-machine
                       systems
60-60.8              Work measurement.  Methods engineering
61-173             Technical education.  Technical schools
173.2-174.5        Technological change
175-178            Industrial research.  Research and
development
```

201-342 Patents. Trademarks
351-385 Mechanical drawing. Engineering graphics
391-995 Exhibitions. Trade shows. World's fairs

TA
1-2040 Engineering (General). Civil engineering (General)
164 Bioengineering
165 Engineering instruments, meters, etc. Industrial instrumentation
166-167 Human engineering
168 Systems engineering
170-171 Environmental engineering
174 Engineering design
177.4-185 Engineering economy
190-194 Management of engineering works
197-198 Engineering meteorology
213-215 Engineering machinery, tools, and implements
329-348 Engineering mathematics. Engineering analysis
349-359 Mechanics of engineering. Applied mechanics
365-367 Acoustics in engineering. Acoustical engineering
401-492 Materials of engineering and construction. Mechanics of materials
495 Disasters and engineering
501-625 Surveying
630-695 Structural engineering (General)
703-712 Engineering geology. Rock mechanics. Soil mechanics. Underground construction
715-787 Earthwork. Foundations
800-820 Tunneling. Tunnels
1001-1280 Transportation engineering
1501-1820 Applied optics. Photonics
2001-2040 Plasma engineering

TC
1-978 Hydraulic engineering
160-181 Technical hydraulics
183-201 General preliminary operations. Dredging. Submarine building
203-380 Harbors and coast protective works. Coastal engineering. Lighthouses
401-506 River, lake, and water-supply engineering (General)
530-537 River protective works. Regulation. Flood control
540-558 Dams. Barrages
601-791 Canals and inland navigation. Waterways
801-978 Irrigation engineering. Reclamation of wasteland. Drainage

TC
1501-1800 Ocean engineering

TD
1-1066 Environmental technology. Sanitary engineering
159-168 Municipal engineering
169-171.8 Environmental protection
172-193.5 Environmental pollution

194-195 Environmental effects of industries and plants
201-500 Water supply for domestic and industrial purposes
419-428 Water pollution
429.5-480.7 Water purification. Water treatment and conditioning. Saline water conversion
481-493 Water distribution systems
511-780 Sewage collection and disposal systems. Sewerage
783-812.5 Municipal refuse. Solid wastes
813-870 Street cleaning. Litter and its removal
878-894 Special types of environment
Including soil pollution, air pollution, noise pollution
895-899 Industrial and factory sanitation
896-899 Industrial and factory wastes
920-934 Rural and farm sanitary engineering
940-949 Low temperature sanitary engineering
1020-1066 Hazardous substances and their disposal
.
.
.
TT
1-999 Handicrafts. Arts and crafts
161-170.7 Manual training. School shops
174-176 Articles for children
180-200 Woodworking. Furniture making. Upholstering
201-203 Lathework. Turning
205-267 Metalworking
300-382.8 Painting. Wood finishing
387-410 Soft home furnishings
490-695 Clothing manufacture. Dressmaking. Tailoring
697-927 Home arts. Homecrafts
Including sewing, embroidery, decorative crafts
950-979 Hairdressing. Beauty culture. Barbers' work
980-999 Laundry work

TX
1-1110 Home economics
301-339 The house
Including arrangement, care, servants
341-641 Nutrition. Foods and food supply
642-840 Cookery
851-885 Dining-room service
901-946.5 Hospitality industry. Hotels, clubs, restaurants, etc. Food service
950-953 Taverns, barrooms, saloons
955-985 Building operation and housekeeping
1100-1105 Mobile home living
1110 Recreational vehicle living