T.C

BAHÇEŞEHİR ÜNİVERSİTESİ

CONCEPT BASED SEMANTIC WEB MINING

Master's Thesis

ALPER ÖZIŞIK

İSTANBUL, 2008

T.C

BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUE OF SCIENCE

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

CONCEPT BASED SEMANTIC WEB MINING

Master's Thesis

ALPER ÖZIŞIK

Supervisor : Assoc. DR. ADEM KARAHOCA

İSTANBUL, 2008

T.C

BAHÇEŞEHİR ÜNİVERSİTESİ

T.C

BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER ENGINEERING PROGRAM

Name of the thesis: CONCEPT BASED SEMANTIC WEB MINING
Name/Last Name of the Student: Alper Özışık
Date of Thesis Defense: 05.09.2008

The thesis has been approved by the Institute of Science and Technology.

Prof.Dr.A.Bülent Özgüler
Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Assoc. Dr. Adem Karahoca
Program Coordinator

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members Title Name and Surname	Signature
Thesis Supervisor: Assoc. Dr. Adem Karahoca	----------
Prof.Dr. Nizamettin Aydın	----------
Asst. Prof. Dr. Yalçın Çekiç	----------

ABSTRACT

CONCEPT BASED SEMANTIC WEB MINING

Özışık, Alper

Master's Degree for Computer Engineering

Supervisor: Assoc. Dr. Adem Karahoca

08, 2008, 42 Pages

Current web search technologies are good to find similar pages with their content and link structures. However they are not enough to find similar pages including word dictionary or cross-linguistic meaning relevance.

This thesis focuses finding similar pages on web with combination of known techniques. Link gatherings, semantic web metadata parsing are required for Web content and structural mining. This thesis differs from other web mining methods with word dictionary meaning and cross-linguistic meanings. All of that information is processed by web crawlers and indexed on data for web mining.

Indexed data is purified from non-useful words and misleading web sites, such as advertisement sites. Clean data is processed in clustering data mining. Data processing contains adding more information to page relations with link distance levels and content word joint values.

For the web mining process, K-means and EM methods of clustering algorithms are compared to decide which one will have better results. Chosen method enlists similar pages to the page of the user selected at starting point of the process.

Keywords: Clustering, Crawler, Cross-linguistic, Resource Description Framework, Semantic Web

ÖZET

KAVRAMA DAYALI ANLAMSAL WEB MADENCİLİĞİ

Özışık, Alper

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

Tez Danışmanı: Doç. Dr. Adem Karhoca

08, 2008, 42 Sayfa

Şimdiki web arama teknolojileri, benzer sayfaları içerikleri ve bağlantı yapıları ile bulma konusunda iyiler. Buna rağmen, benzer sayfaları sözlük kelime ve çapraz dil karşılıklarının alakadarlıklarını bulma konusunda iyi değiller.

Bu tez benzer sayfaların bulunmasına bilinen yöntemlerin kombinasyonu ile yoğunlaşıyor. Bağlantı toplama, anlamsal tanımlayıcı veri algılanması web içerik ve yapısal madenciliği için gereklidir. Bu tez, diğer web madenciliği tekniklerinden sözlük anlamları ve çapraz dildeki anlamlarını da içererek ayrılıyor. Web robotları tarafından toplanan tüm bu veriler, web madenciliği için veri tabanında dizinlenir.

Dizinlenmiş veri, içindeki anlamsız kelimelerden ve yanlış yönlendirici sitelerden, mesela reklam sitelerinden, arındırılır. Temiz veri kümeleme veri madenciliği için işlenir. Bu işleme sırasında, sayfa ilişkilerine sayfa bağlantı seviye bilgisi ve içeriklerindeki kelimelerin kesişim değerlerini eklenir.

Web madenciliği işlemi için, kümeleme algoritmalarının K-means ve EM metotları, hangisi daha iyi sonuç verecek diye karşılaştırıldı. Seçilen metot, kullanıcının başta seçmiş olduğu sayfa ile benzer sayfaları listeledi.

Anahtar Kelimeler: Anlamsal Örün, İnternet Robotu, Kaynak Tanımlama Çerçevesi, Kümeleme

TABLE of CONTENTS

LIST of TABLES

LIST of GRAPHICS

LIST of CODES

LIST of ABBREVATIONS

8-bit UCS/Unicode Transformation Format	UTF 8
A computer programming system created by Donald Knuth to implement literate programming	WEB
Application Programming Interface	API
Common Gateway Interface	CGI
Common Language Runtime	CLR
Data Flow Diagram	DFD
Data Mining Extensions	DMX
Data Source	DS
Data Source View	DSV
Dublin Core	DC
Extensible Markup Language	Xml
HTML Anchor Element	a
HyperText Markup Language	HTML
HyperText Transfer Protocol	HTTP
Open Directory Project	ODP
Random access memory	RAM
Resource Description Framework	RDF
Simple Object Accsess Protocol	SOAP

Uniform Resource Identifier	URI
Uniform Resource Locator	URL
Web Ontology Language	OWL
World Wide Web. Invented in 1989 by Sir Tim Berners-Lee, -- a hypertext system that operates over the Internet, used for serving Web pages and transferring files	WWW

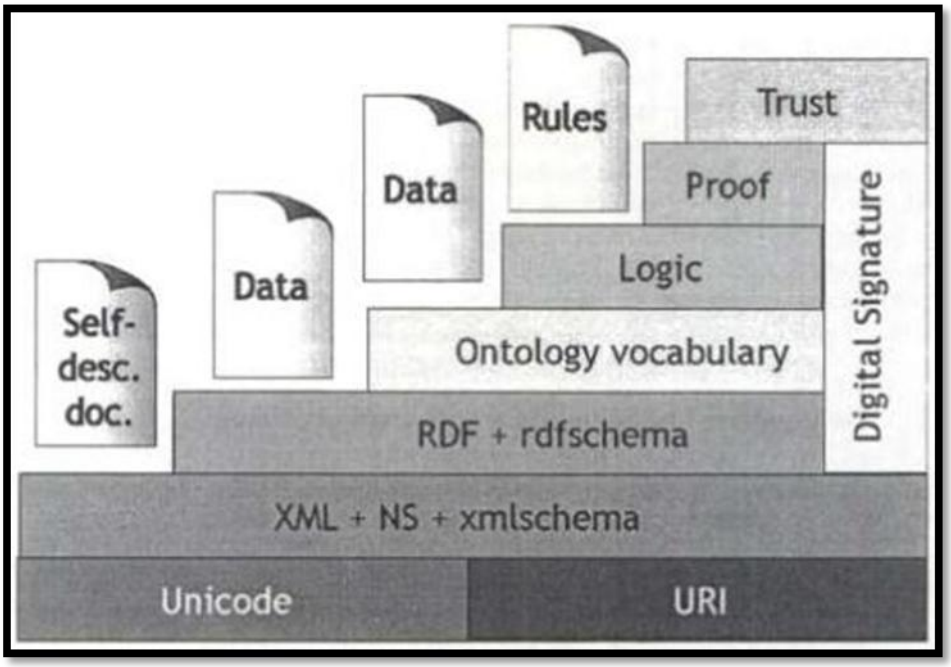
1. INTRODUCTION

1.1 SEMANTIC WEB

Semantic web aims to convert unstructured (huge) data to Human and Machine understandable data. That information can be mined to extract useful information.

Gerd Stummea, Andreas Hotho, Bettina Berendt, 2006, pages 125-126:

> *Today's search engines are quite powerful, but still too often return excessively large or inadequate list of hints. Machine processable information can point the search engine to the relevant pages and can thus improve both precision and recall.*
>
> *It is almost impossible to retrieve information with a keyword search when the information is spread over several pages.*
>
> ***Directions of semantic web is heading:***
>
> *1. Providing a common syntax for machine understandable statements.*
>
> *2. Establishing common vocabularies.*
>
> *3. Agreeing on a logical language.*
>
> *4. Using the language for exchanging proofs.*



Source: P. Patel-Schneider, D. Fensel (2002), Layering the semantic Web: Problems and directions

Graphic 1.1 : Layers of the Semantic Web

On the first two layers, a common syntax is provided. Uniform resource identifiers (URIs) provide a standard way to refer to entities. The Extensible Markup Language (XML) fixes a notation for describing labeled trees, and XML Schema allows the definition of grammars for valid XML documents. XML documents can refer to different namespaces to make explicit the context of different tags. The Resource Description Framework (RDF) can be seen as the first layer where information becomes machine understandable.

1.1.1 RDF model

Gerd Stummea, Andreas Hotho, Bettina Berendt, 2006, page 126:

RDF documents consist of three types of entities: Resources, properties, and statements. Resources may be Web pages, parts or collections of Web pages, or any (real-world) objects which are not directly part of the WWW. In RDF, resources are always addressed by URIs. Properties are specific attributes, characteristics, or relations describing resources. A resource together with a property having a value for that resource form an RDF statement. A value is either a literal, a resource, or another statement. Statements can thus be considered as object-attribute value triples.
The data model underlying RDF is basically a directed labeled graph. RDF Schema defines a simple modeling language on top of RDF which includes classes, is-a relationships between classes and between properties, and domain/range restrictions for properties. RDF and RDF Schema are written in XML syntax, but they do not employ the tree semantics of XML.

1.1.2 Semantic Web Mining

Gerd Stummea, Andreas Hotho, Bettina Berendt, 2006, pages 128-129:

Web Mining is the application of data mining techniques to the content, structure, and usage ofWeb resources.
Web content mining analyzes the content of Web resources. Today, it is mostly a form of text mining.
In addition to standard text mining techniques, Web content mining can take advantage of the semi-structured nature ofWeb page text. HTML tags and XML markup carry information that concerns not only layout, but also logical structure.
Web structure mining usually operates on the hyperlink structure of Web pages. Mining focuses on sets of pages, ranging from a single Web site to the Web as a whole. Web structure mining exploits the additional information that is (often implicitly) contained in the structure of hypertext.
In Web usage mining, mining focuses on records of the requests made by visitors to a Web site, most often collected in a Web server log.

1.2 WHAT IS WEB SEARCH?

Typically web search is a keyword based web page finding service. User enters keywords or other search parameters to find a website that fills his/her needs. There are many web search engines on the web like Google, live.com, Yahoo!, etc. For example,

when a user makes query on Google, it shows results only which are found according to keyword based search. However, it may not be possible to find web pages which use the same meaning of those keywords which are on the searched web page. Keyword based searches do not search the web pages based on web semantic properties.

Semantic web is an evolving extension of World Wide Web. Its purpose is to add or extract information from human readable data to machine understanding. There are two technologies endorsed by World Wide Web Consortium. Web Ontology Language (OWL) is a family of knowledge representation for authoring ontologies. OWL has three sub languages: OWL Lite, OWL DL and OWL Full. OWL Lite and OWL DL are based on Description Logics. OWL Full uses a novel semantic model, which provides compatibility with Resource Description Framework (RDF). RDF is a metadata modeling language, which is based on XML syntax.

Web semantic search is obtaining results through the semantic metadata properties of web resources, such as keywords, author or description. To provide semantic data a parser must be applied to web pages. There is a standard of RDF schema defined by Dublin Core (DC) to categorize properties of web resources, including web pages, images, music and video files. Defined properties of web resources by Dublin Core are: Title, Creator (author), Subject or Keywords, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights Management. Web semantic search is finding specific property value of web resource.

Web keyword search, searches the web for the exact word or the word that sounds like to it. However it does not retrieve the web pages that are using same meaning of the word. The information that user really needs could be on the other page which has the same meaning of the keyword. This type of search may increase efficiency of the results while searching information through similar but from different sources.

Current web search solutions, which are using multi-culture features, do not perform search the word with its translation on selected languages. If user is capable of using different languages, cross-language dictionary meaning results will provide better results and speeds up the user.

Web pages, that are similar, do have entropy on their keywords, titles, description and even might be from the same author. In order to understand pages with similar properties, their links and their metadata properties must be clustered. Clustering is unsupervised grouping of data. It is done by machine with the criterions of their links and values. Web page clustering will result in web page clusters; in each cluster similar pages are grouped.

1.3 OTHER PROJECTS THOSE ARE SIMILAR TO THIS ONE

1.3.1 Enhancing the power of the internet using fuzzy logic-based web intelligence: Beyond the semantic web

With their ease of use World Wide Web search engines are the most consumed online services. Semantic web has many differences from World Wide Web. Since Semantic web mainly provides a common framework that allows data to be shared across applications which is then shared by community through World Wide Web.

This paper's purpose has been to go beyond traditional semantic web, which has been defined and stored as mesh or databases within the World Wide Web, with the view of "Before one can use the power of semantic web, the relevant information has to be mined through the search mechanism and logical reasoning". In this document two main motivations have been addressed for a new intelligent search engine.

1) As the web environment is unstructured and imprecise to deal with information there is a need for a logic that includes modes of reasoning for to extract approximate data other than exact data.

2) Searches may have many results. Finding decision-relevant and query relevant information in imprecise environment is the challenging problem.

In this document, a database and a decision model has been explained. A framework to extract the information from web sites and store them is to be used in search engines (Nikravesh, M., 2007).

1.3.2 An interactive agent-based system for concept-based web search

Generally users are not satisfied with the results retrieved as there are missing facts like specifying appropriate queries and keyword-based similarity ranking presently encountered by search engines. To overcome these issues this document is presenting a multi-agent framework which is also using the feedbacks gathered from the users as he identifies the pages related to his/her search. As the system gathers those feedbacks and use those data while processing to formulate the queries. This document indicates that the experimental results are showing the effectiveness of this framework so that the concept-based semantic search could be achieved (Lee, W.P. & Tsai,T.C., 2003).

1.3.3 Improving web-query processing through semantic knowledge

(Conesa,J., Storey,V.C. and Sugumaran,V., 2007):

> *This document again points to the problem of obtaining irrelevant solutions from the searches. For that reason different types of knowledge are used for querying of success. This document is presenting a methodology for processing web queries that includes semantic knowledge from different application domains like ResearchCyc, WordNet. An analysis of different queries from different application domains using the semantic and linguistic knowledge illustrates how more relevant results can be obtained.*

1.3.4 Regularized query classification using search click information

Hundreds of millions of users each day submit queries to the Web search engine. The user queries are typically very short which makes query understanding a challenging problem. In this paper, we propose a novel approach for query representation and classification. By submitting the query to a web search engine, the query can be represented as a set of terms found on the web pages returned by search engine. In this way, each query can be considered as a point in high-dimensional space and standard classification algorithms such as regression can be applied. However, traditional regression is too flexible in situations with large numbers of highly correlated predictor variables. It may suffer from the over fitting problem. By using search click information, the semantic relationship between queries can be incorporated into the

learning system as a regularizer. Specifically, from all the functions which minimize the empirical loss on the labeled queries, we select the one which best preserves the semantic relationship between queries. We present experimental evidence suggesting that the regularized regression algorithm is able to use search click information effectively for query classification (He,X. and Jhala, P., 2008).

1.3.5 SemreX: Efficient search in a semantic overlay for literature retrieval

The speed of sharing www is so enormous that the search engines can manage only a small part of it. It is an effective information sharing model though content searching still remains a serious challenge of large scale peer-to-peer networks. In this paper semantically similar peers are locally clustered together and long-range connections are rewired for a short-cut in peer-to-peer Networks. Based on this semantic overlay, a heuristic query routing algorithm is proposed for efficient content searching (Jin,H. and Chen, H., 2007).

1.3.6 Symbolic links in the Open Directory Project

They present a study to develop an improved understanding of symbolic links in web directories. A *symbolic link* is a hyperlink which makes a directed connection from a webpage along one path through a directory to a page along another path. While symbolic links are ubiquitous in web directories such as Yahoo!, they are under-studied and, as a result, their uses are poorly understood. A cursory analysis of symbolic links reveals multiple uses: to provide navigational shortcuts deeper into a directory, backlinks to more general categories, and multiclassification. They have investigated these uses in the Open Directory Project (ODP), the largest, most comprehensive, and most widely distributed human-compiled taxonomy of links to websites, which makes extensive use of symbolic links. The results reveal that while symbolic links in ODP are used primarily for multiclassification, only few multiclassification links actually span top- and second-level categories. This indicates that most symbolic links in ODP are used to create multiclassification between topics which are nested more than two levels deep and suggests that there may be multiple uses of multiclassification links. They also situate symbolic links *vis à vis* other semantic and structural link types from hypermedia (Perugini, S.,2007).

1.4 PROPOSED APPROACH

To be able to find similar web pages, every web page must be categorized by its different properties, mainly its metadata. The metadata has been read by a crawler then processed and indexed. This metadata can have relationships with other words. So word-metadata relation must exist in the database. This relation can also include cross-language meanings of the word.

The web page metadata + word & metadata relation + word meanings are not just enough to get efficient web page similarity information. Web page links also must be considered.

Accounting all data in a model, clustering those web sites, to get which URL fits into a cluster will give us similar page results.

2. MATERIALS AND METHODS

2.1 MATERIALS

To collect RDF data from web pages, RDF parser (DC-dot) has been executed that has Dublin Core RDF schema. Dublin Core is one of the most mature RDF schemas to describe web resources. More information about DC-dot can be found on http://www.ukoln.ac.uk/metadata/dcdot/help/cgi-params.html . More information about Dublin Core can be found on http://www.dublincore.org/ . A parser for RDF/XML has been developed for this RDF generator.

For dictionary meanings, SesliSozluk is picked, because it is an online cross-language dictionary, which supports primarily Turkish and English. It gives lesser support for German. More information can be found on http://www.seslisozluk.com/ . A HTML parser has been developed for this web site.

For finding roots of Turkish words, Turkish Lexical Database Project on Sabancı University has been used. More information can be found on http://www.hlst.sabanciuniv.edu/TL/ . A HTML parser for this web page has been developed.

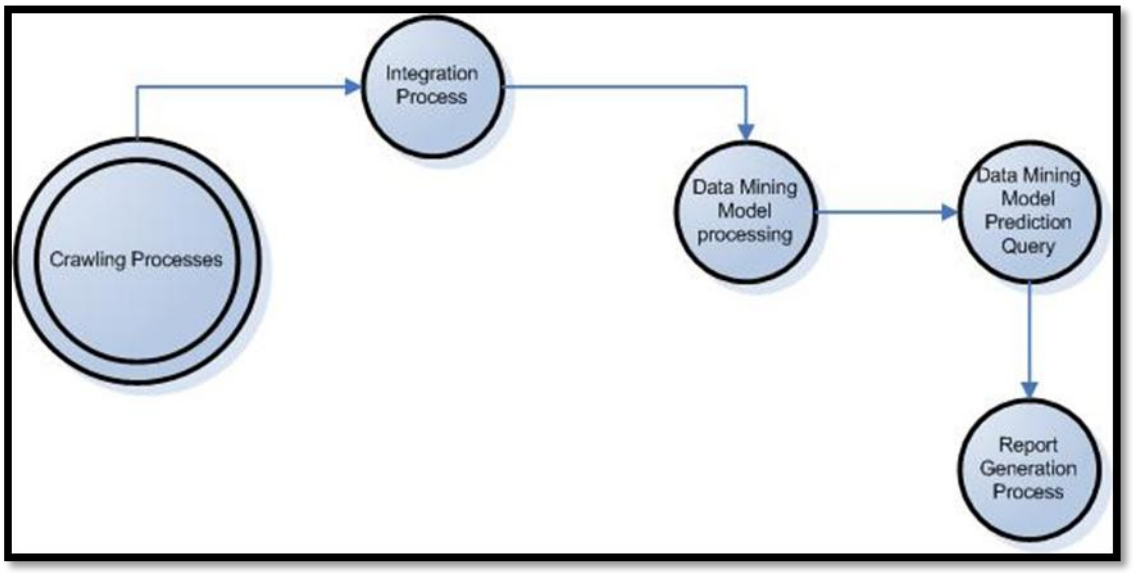
For making a keyword search, URL table must be filled. SOAP API of Live™ had been used. A developed function gathers URLs from search results up to certain number of results. The function is based on a sample provided on http://www.microsoft.com/downloads/details.aspx?FamilyID=E32DAC6F-ABFC-4C07-9AA3-4EF95883883D&displaylang=en .

Microsoft .net is one of the world leading technologies. Microsoft .net has been selected as development platform and one of its programming languages, C# is used in this project.

Microsoft SQL Server 2008 is one of the world most performing data base server and it has data mining features. SQL Server 2008 is a companion data base server application to Microsoft.net 3.5 platform.

On Microsoft SQL Server 2008 data mining applications are developed with Microsoft Visual Studio 2008. In Analysis Services project, data mining models can be applied to a data source.

2.2 PROJECT FLOW



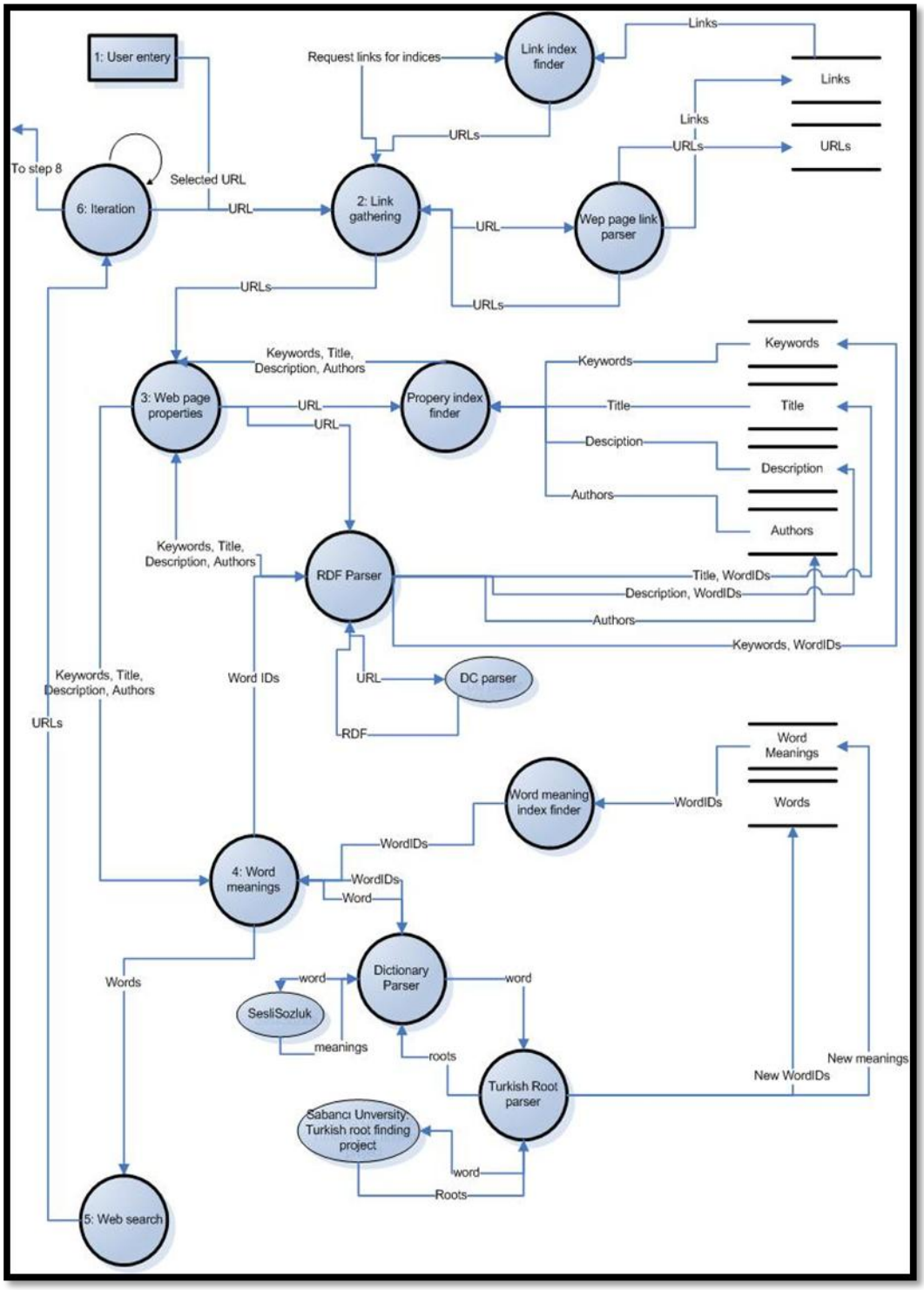
Graphic 2.1 : Data mining process flow

To solve that approach, for grouping similar pages can be basically explained by 6 tasks: Parsing web page for **links**, gathering **RDF** schema, finding **meanings**, **indexing**, **processing** and **integration**, **clustering**. The application will run in following order:

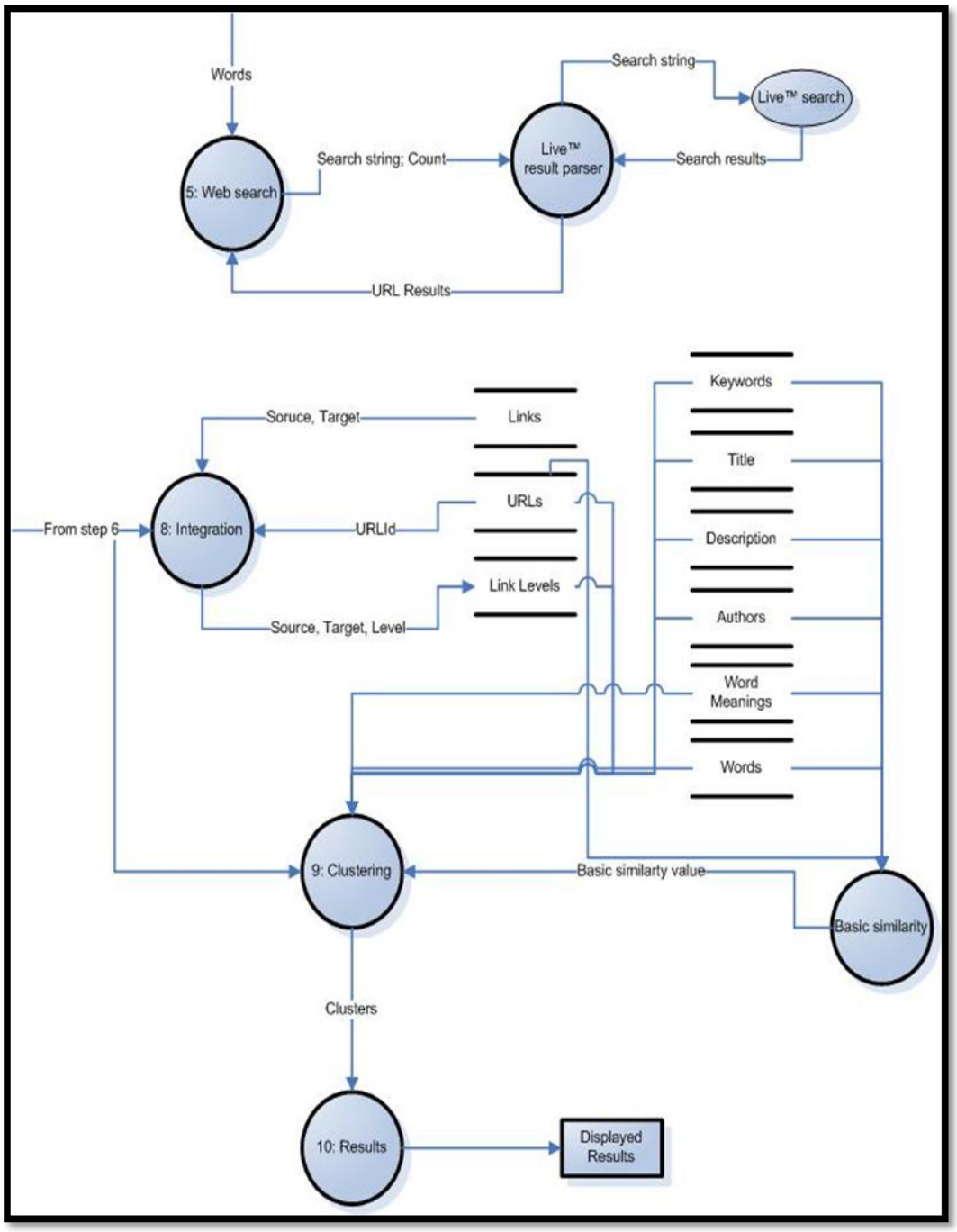
1. User selects a web page. In all cases, if it has some indexed data; indexed data will be used for speeding up the process instead of parsing.
2. Parsing web page for links.
3. RDF schema for links and page will be parsed.

4. Dictionary meanings and cross-language meanings of words will be found in RDF schema.
    a. If no result for meaning, a Turkish word root finding method, word will be searched again in dictionary.
5. A keyword based search will run on current web search engines to find some similar pages.
6. For each result in step 5, steps 2 through 4 will be applied.
7. Each gathered new information will be processed and cleaned for some noises in data while indexing in steps 2 to 4.
8. To make it use for clustering method, some data must be extracted. This is the integration process.
9. Data will be clustered.
10. Clustered data will be displayed, showing the selected page, the pages belong to same cluster will be displayed to user.

The graphics below are the **Data Flow Diagram for the generics of the process:**



Graphic 2.2: Main data flow diagram Part A



Graphic 2.3 : Main data flow diagram Part B

2.3 DATABASE MODEL AND PROGRAMMING

In this section, SQL Server Database schema and some procedures that are used in application will be explained.

2.3.1 Database Schema

By designing the database, normalization rules are considered. Every data represented with a unique by table data, which is the primary key of the table.

Below is the list of tables used in database:

```
Urls
        URLId           int     primary key     auto number
        URL             nvarchar (2000)
        WebSiteID       int

WebSites
        WebSıteID       int     primary key     auto number
        WebSite         nvarchar (500)
        Useful          bit

Links
        SoureURLId      int     primary key
        TargetURLId     int     primary key

Authors
        AuthorID        int     primary key     auto number
        AuthorName      nvarchar (100)

URLAuthors
        URLId           int     primary key
        AuthorID        int     primary key

Words
        WordID          int     primary key     auto number
        Word            nvarchar (500)
        Useful          bit

Word2Word
        Word1ID         int     primary key
        Word2ID         int     primary key
```

Code 2.1 : Database tables

```
URLKeywords
        URLId           int     primary key
        WordID          int     primary key

URLTitle
        URLId           int     primary key
        WordID          int     primary key

URLDescription
        URLId           int     primary key
        WordID          int     primary key
```

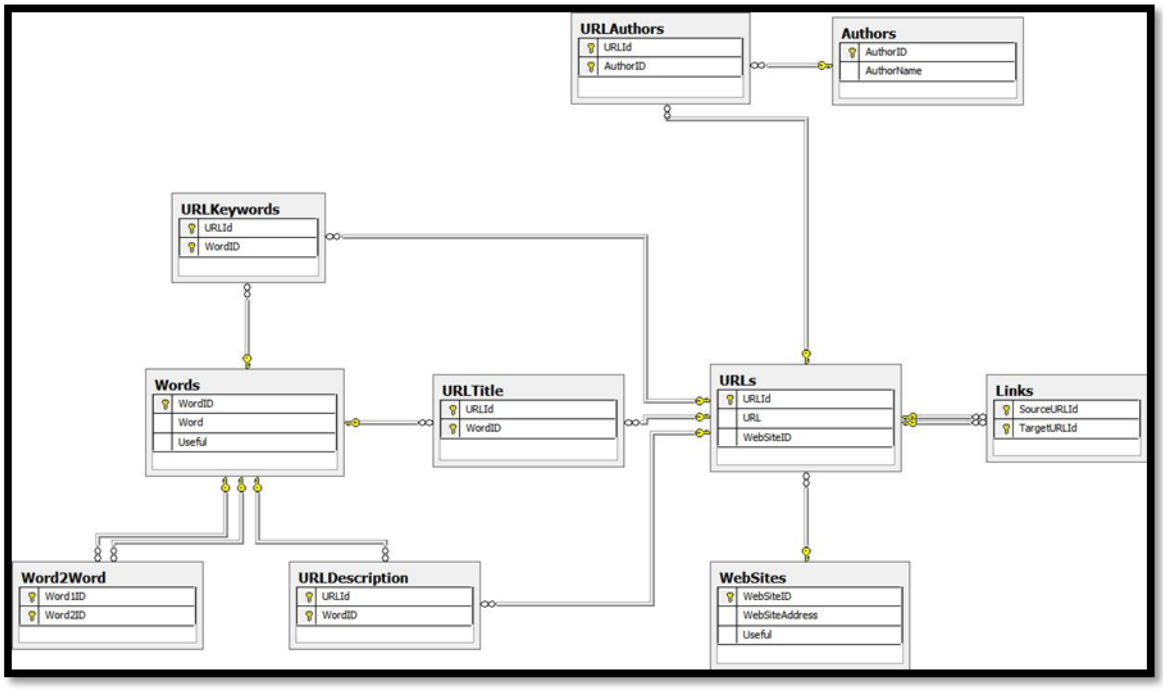
Code 2.1 : Database Tables (cont.)

Keywords, title and descriptions can be represented by words. Every word has an ID, which is stored in Words table. Thus the URLKeywords, URLTitle and URLDescription tables holds WordID instead of word.

URLAuthors table stands for, if the web page has multiple authors.

WebSites table is to store additional information of the host. WebSiteAddress column is derived from URL, the host value of the URI, while querying database when adding new website, holding the column WebSiteAddress will fastens the execution time of the adding process. Useful column of the WebSites table is to mark that website will be considered for parsing links from it and clearing data for data mining.

Where the word is stored is words table, other tables do not store any word value. Useful column stands for representing that this word has any meaning for deciding that web page's actually about is. Those words can be exampled as: "**and or for to etc**"



Graphic 2.4 : Entity relationship diagram

In addition those related tables, some several tables are cached results of database stored procedures and data mining processes.

```
LinkLevels
        SourceURLId     int
        TargetURLId     int
        level           int

ClusterResults
        URL             nvarchar(255)
        $CLUSTER        nvarchar(255)

ClusterWord
        Cluster         nvarchar(255)
        Word            nvarchar(50)
```

Code 2.2 : Database helper tables

1. LinkLevel table is derived from Links table. It is full cross query result of URLs. Every URLId linked with other URLId. Level column identifies how many paths must be traversed to reach from source to target URL. A stored procedure in database processes this table, deletes first and then fills it with levels.

2. ClusterResults table created by Analysis Services, it is the prediction query result of clustering data mining model.

3. ClusterWord table is a cache table of a query. It takes long time to execute the query, result of it saved as a table. This table will be used for post data mining query analysis, while getting Word versus Cluster decisions.

In database some views are for data mining models. Those are:

```
vURLDescriptionWords
        URLId           int
        WordID          int

vURLKeywordsWords
        URLId           int
        WordID          int

vURLTitleWords
        URLId           int
        WordID          int

vUSefulWords
        WordID          int
        Word            nvarchar (50)

vUsefulURLs
        URLId           int
        URL             nvarchar (2000)

vAdvancedLinks
        SourceURLId     int
        TargetURLId     int
        LinkEntrophy    float
        BasicSimilarity float

vAdvancedLinks2
        rownum          int
        SourceURLId     int
        TargetURLId     int
        LinkEntrophy    float
        BasicSimilarity float
```

Code 2.3 : Views

1. vURLDescriptionWords, vURLTitleWords, vURLKeywordWords are views to display more WordID with the meaning of them excluding non useful words.

2. vUsefulWords displays only useful of them.

3. vUsefulURLs displays URLs with useful websites.

4. vAdvancedLinks works similar as Links table. It contains additional two fields. LinkEntropyh is a similarity value of links based on Link Levels. BasicSimilarity is for displaying similarity of metadata values based on a function explained in database programming section. (GetBasicSimilarty)

5. vAdvancedLinks2 displays an additional column, which is the row number of the view. This column is used for data mining model key for links.

2.3.2 Database Programming

There are several stored procedures and user defined functions written with CLR on database to provide some methods while adding data to database. Those are:

1. **GetAuthorIDWithCreation:** *Stored Procedure;* Program is about to add an author data to database. It checks the database for an existing author data for that is about to being added. If exists, it returns the existing AuthorID, else it adds the data to database and returns new AuthorID of the data that has been added.
2. **GetWordIDWithCreation:** *Stored Procedure;* Program is about to add a word data to database. It checks the database for an existing word data for that is about to being added. If exists, it returns the existing WordID, else it adds the data to database and returns new WordID of the data that has been added.
3. **spAddLinks:** *Stored Procedure;* Program is about to add links of an URL. Links contains a list of URLs. Before adding any URL to database, it checks its existence in table, if exists, it reuses its URLId during link adding. If URL does not exist on table, it adds data then it uses the new URLId during link adding. Every SourceURLId and TargetURLId must exist on URL table, which is

provided also relation foreign keys. Adding to URLs table supplies new URLId for new URLs. Then they can be added to Links table.

4. **HasAnyDublinCoreValue:** *User Defined Function;* It takes an URL string as a parameter. It returns a boolean value of the URL had and metadata information added earlier in database. It checks in order the following tables: URLKeywords, URLTitle, URLDescription, URLAuthors. This order is determined by developer to get performance while querying, because this order shows whether the data most likely exists in.
5. **GetSingleDepthLinkLevel:** *User Defined Function;* It calculates how many links must be traversed to reach from source URLId to target URLId. It uses shortest path methods which has been optimized for this database.
6. **spRefreshData:** *Stored Procedure;* With use of GetSingleDepthLinkLevel, it calculates link levels of each link path with custom iterative method which is similar to shortest path algorithm of Floyd–Warshall. This procedure clears first LinkLevel table and then saves all results into LinkLevels.
7. **GetBasicSimilarty:** *User Defined Function;* It calculates entropy between two URLs. To make a mean value calculation, it uses a function based on intersection of words of URLs in each different metadata field.

2.4 INDEXING METHODS

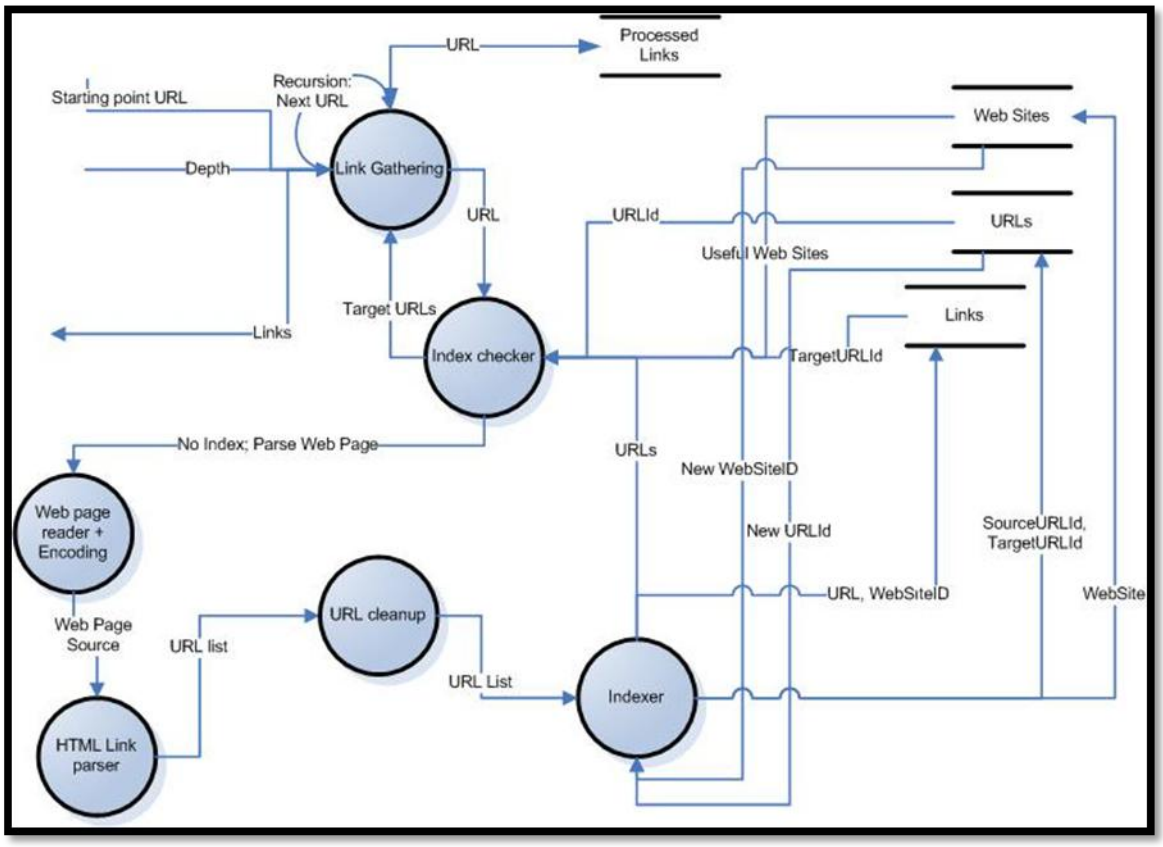
2.4.1 Link Gathering

A web page has links. Those links leads to other related web pages, which must be considered while finding relations between web pages. A generic HTML parser has developed for parsing HTML element **a**.

Every page can have a different encoding. While parsing a web resource, encoding of web page must be considered. This parser also detects web page encoding. If detection fails, parser uses UTF8 as default encoding.

Before parsing a web page, parser checks database for signs of earlier link gathering process for web page. If parser sees a link in database, it uses that information to return the caller. Otherwise it is its parsing job and generates links from web page source.

Not every link is meaningful. Such as advertisement links. They might lead to a relative product; however the URL changes every time, making it impossible to index, because those links might be runtime generated temporary URLs. While reusing indexed data, host of the URL must be considered. Therefore, a column stands on Web Sites table, named as Useful. Known advertisement sites can be marked as Useful = 0.



Graphic 2.5 : Link gathering detailed DFD

The DFD above explains how the Link gathering actually works in detail. Some parts must be also described by text:

1. Processed Links data store is not a database table. It is a program variable list. It tells which links have been processed this time, so it will not be processed again in this time in case of recursive links occur.

2. Index checker process checks links data table for an existing link. However it also checks target URL as if it is useful from URLs table and WebSites table. If useful it returns links. If not useful, returns an empty array.
3. Indexer adds new URLs to URLs data table. During adding process it does not know the new URLId. It selects the identity value of the table to get URLId. Same applies for the WebSites data table. WebSiteID is unknown to process until identity returned from database.
4. Decision to add new WebSite, is made by Indexer process, it queries database against an existing host value of URL.
5. URL cleanup, removes some confusing elements in URL, such as bookmark tags (#). If link do not have host value, it adds the relative host and path value of the source URL including protocol type.
6. Link from Indexer to Index finder sends URLs without needing Index finder to query database to get URLs.

2.4.2 Metadata Extraction: RDF (Dublin Core) Parser

To query DC-dot program, the developer of the project has added feature and its documentation how to use DC-dot with CGI parameters.

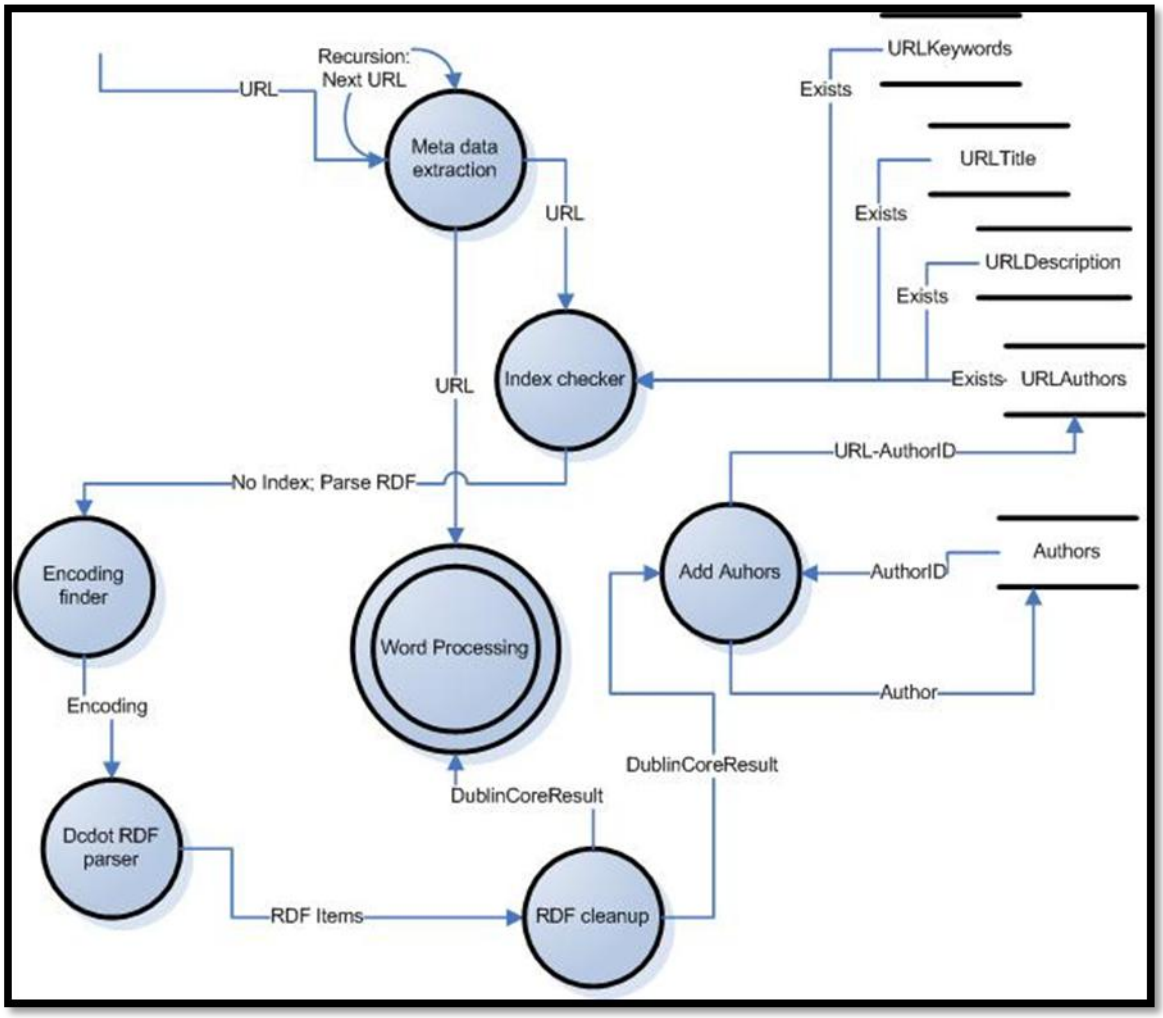
First HTTP request must be made web page, not to DC-dot, to get web page encoding setting, because the result of DC-dot might not be readable to us with different encoding. To get RDF result from web service, HTTP request will made with a query string that includes the web page URL. Result is a RDF, it can be consumed easily with Microsoft.net XmlDocument class. XmlDocument class offers GetElementsByTagName method, which ease the retrieval of values.

Those values generated by DC-dot must be post-processed to get more items. Some items are separated with semicolons. The example below is a RDF result generated by DC-dot on http://www.dotnetnuke.com/ page:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF SYSTEM
"http://dublincore.org/documents/2002/07/31/dcmes-xml/dcmes-xml-
dtd.dtd">
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.dotnetnuke.com/">
    <dc:title>
      DotNetNuke / DNN - Home (4.9.0 RC1)
    </dc:title>
    <dc:creator>
      DotNetNuke
    </dc:creator>
    <dc:subject>
      DotNetNuke; web application framework; NetNuke; .NetNuke;
      dot netnuke; dot net nuke; DNN; DDN; IBuySpy; IBS; open
      source; OS; content management system; CMS; cms; asp.net;
      ASP.NET; ASP; Visual Studio; VS; .NET; dot net; VS.NET;
      site builder; blog; gallery; forums; chat; survey; windows;
      server; free; download; community; collaboration; portal;
      sharepoint; alfresco; ruby on rails; ror; liferay; spring;
      zope; drupal; plone; xoops; mambo; nuke; ECM; WCM; module;
      skin; support; DotNetNuke; DNN
    </dc:subject>
    <dc:description>
      DotNetNuke is an open source web application framework
      ideal for creating, deploying and managing interactive web,
      intranet, and extranet sites securely.
    </dc:description>
    <dc:publisher>
    </dc:publisher>
    <dc:type>
      Text
    </dc:type>
    <dc:format>
      text/html; charset=utf-8
    </dc:format>
    <dc:format>
      77754 bytes
    </dc:format>
  </rdf:Description>
</rdf:RDF>
```

Code 2.4 : Example of Dublin Core RDF

Parser method returns a DublinCoreResult class which is developed by me. This class has array of word values in Title, Description and Keywords (subject) and string values of authors (creator). To get word values from parser class, all strings are split to words and stored at corresponding result store type.



Graphic 2.6 : Metadata extraction detailed DFD

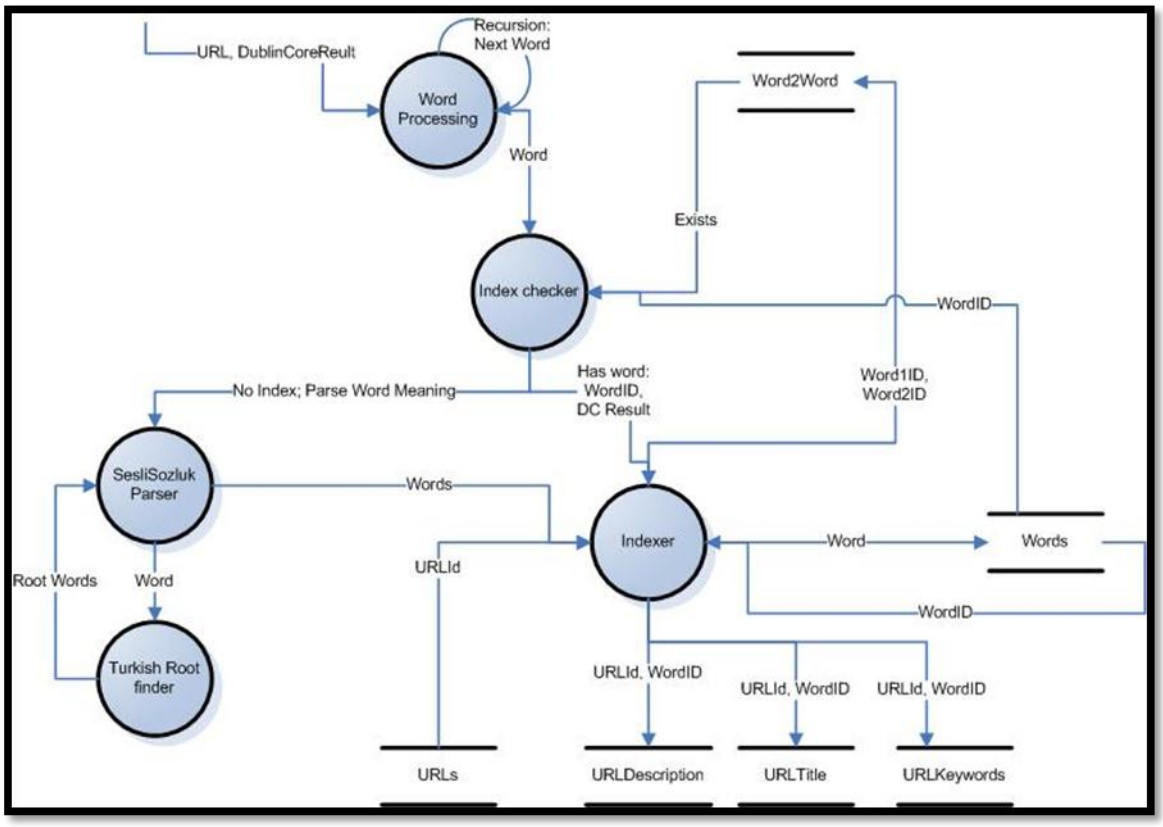
The DFD above shows how RDF parser actually works in detail. Some proportion of it must be explained in text:

1. Index checker only checks if URL has **any** information before on database. It checks in order 4 tables: URLKeywords, URLTitle, URLDescription, URLAuthors. This process does not return RDF result to anywhere. Only decides if the page must be parsed.

2. After RDF cleanup process, resultant class sent to Word Processing, which will be defined in next section.
3. This whole metadata extraction process do only writes only Author information to database. Each other data needs words to be processed before saving on database.
4. If database contains any metadata information on URL, program does not enter to Word Processing step, because if it has that information, worlds are already pre-processed.

2.4.3 Word Processing

Word processing method is for checking whether the word is indexed, if indexed and has meaning, uses them for Title, Keyword, Description mapping of the URL. If not indexed, finds the meanings then indexes it for use in Title, Keyword, Description mapping. In some cases, Turkish words might not be understood by dictionary. In those cases roots of the Turkish words must be found.



Graphic 2.7 : Word Processing Detailed Data Flow Diagram

Above is the detailed Data Flow Diagram of the Word Processing

1. Index checker, checks Word2Word, word meaning table for the word, that it has any meanings. If word has any meanings, it passes the WordID and Dublin Core Result to indexer. If it has no meaning, SesliSozluk Parser begins to look for meanings of the word.
2. SesliSozluk parser parses the web page on SesliSozluk.com. It takes unique Turkish and English words as meaning of the word. If no result has been found it transfers the word to Turkish Root finder process, for it might be a Turkish word that had suffixes. This action is done once for every word, to prevent endless loops.
3. Turkish Root parser parses the web page on Sabancı University by sending program generated web form post requests. It might find more than one different root word for the word that had been searched. Programs could not determine which one is the correct one, because program does not make language based

sentence meaning processing. It passes found root words to SesliSozlukParser to be searched in dictionary.

4. SesliSozluk parser passes found words to indexer.
5. Indexer takes Words and Dublin Core Result to store metadata data of the web page. Each properly of Dublin Core Result had stored on database by corresponding table by WordIDs.

2.4.4 Web Search Engine & Crawlers

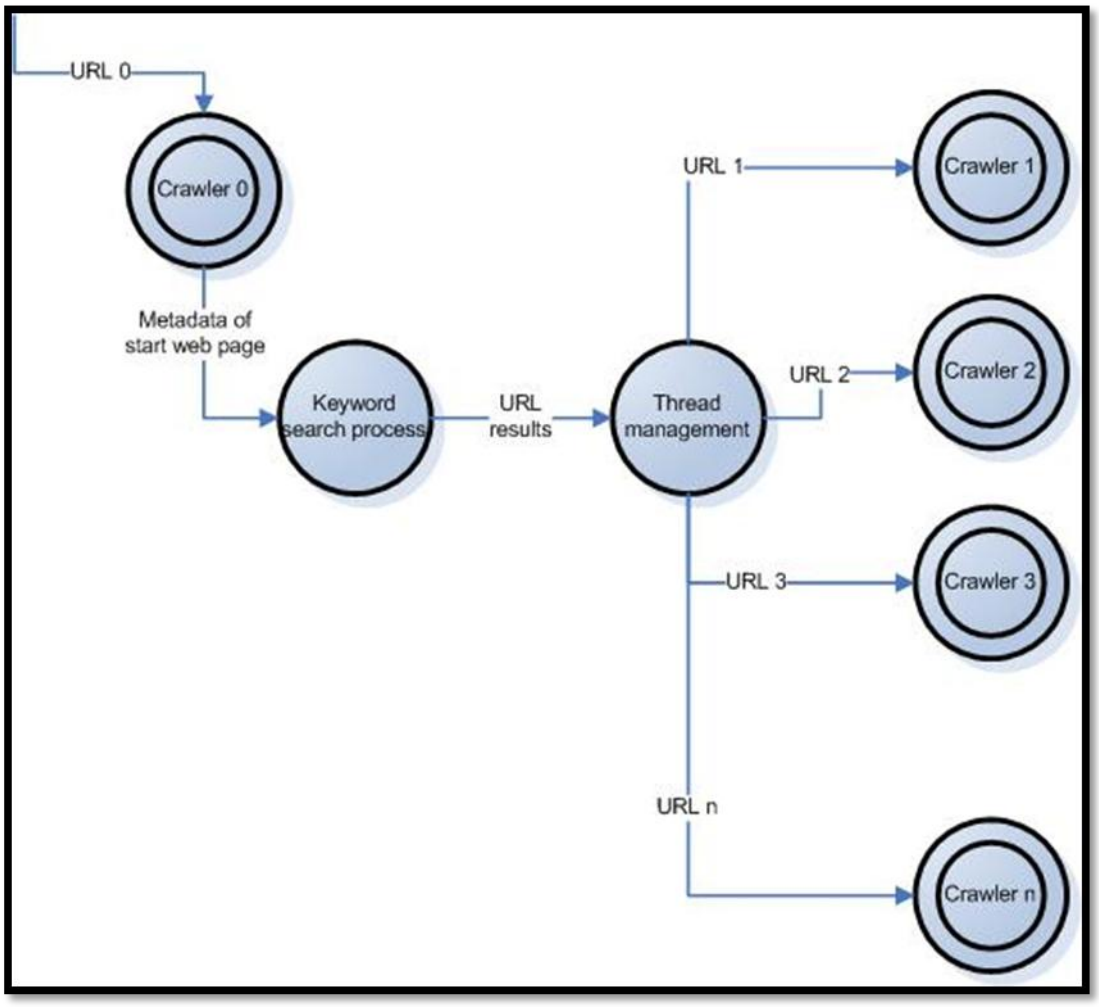
Let's call all the previous steps as crawler process. This process is Crawler 0. After Crawler 0 finishes its job, metadata of the web page (URL 0) and its linked pages with their metadata had been added to database. Keyword search process begins with gathering metadata search string from database to make a web search. All distinct words and authors contacted in quotes with OR operator.

In keyword search process Live™ SOAP API had been used. While querying through that API, it pages the results, 50 maximum in one time. With an offset definition, more results can be parsed sequentially or simultaneous.

Results of keyword based web search are an entry point for indexing more pages. Each URL passed to thread management process. This process splits jobs to crawlers. Each crawler does same jobs and iterations same as Crawler 0 with different starting point. One crawler is responsible for one starting point.

There is a common processed URLs list on application. When crawler finishes gathering links from it, the URL has been added there. During crawling process, if the current URL to be processed became one of the URL in the list, crawler skips it, because it has been processed earlier. This also prevents some extra recursions while crawling.

Thread management waits all crawler processes to finish their jobs, until then it suspends its self. Each crawling process, checks whether it is the last thread in the list. Last finishing crawling process, resumes the Thread Management thread and indexing processes are finished.

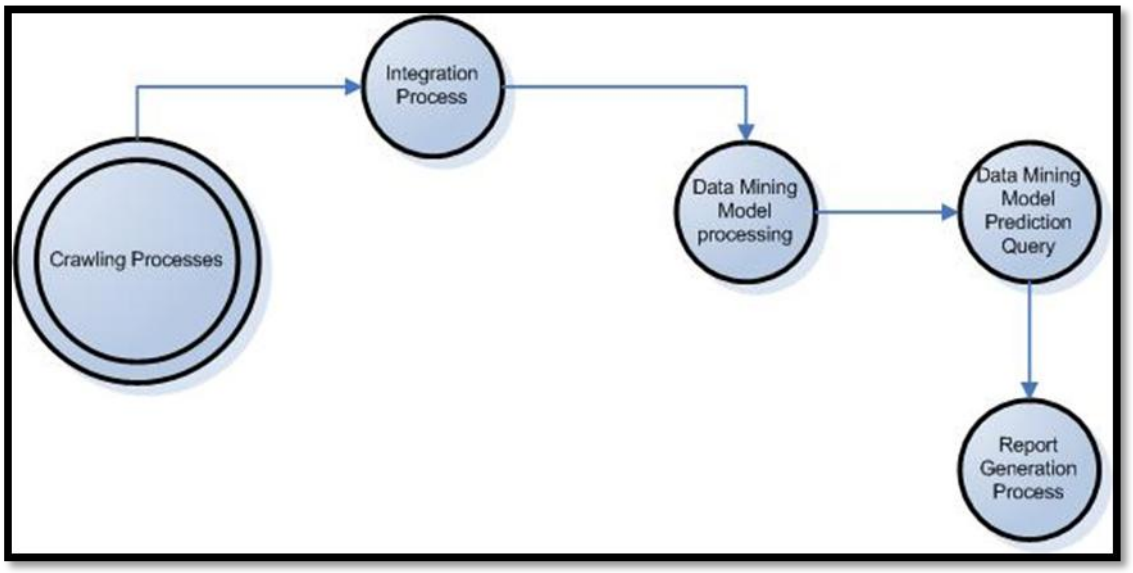


Graphic 2.8 : Crawlers detailed DFD

Above is the detailed Data Flow Diagram of web keyword based search and crawler threading.

2.5 DATA MINING METHODS

SQL Server 2008 Analysis Services is picked for data mining platform. To develop a data mining model, a Visual Studio 2008 Business Intelligence Project, Analysis Services Project must be created. In following sections what those steps are and how the data mining model was queried will be explained.



Graphic 2.9 : Data mining process flow

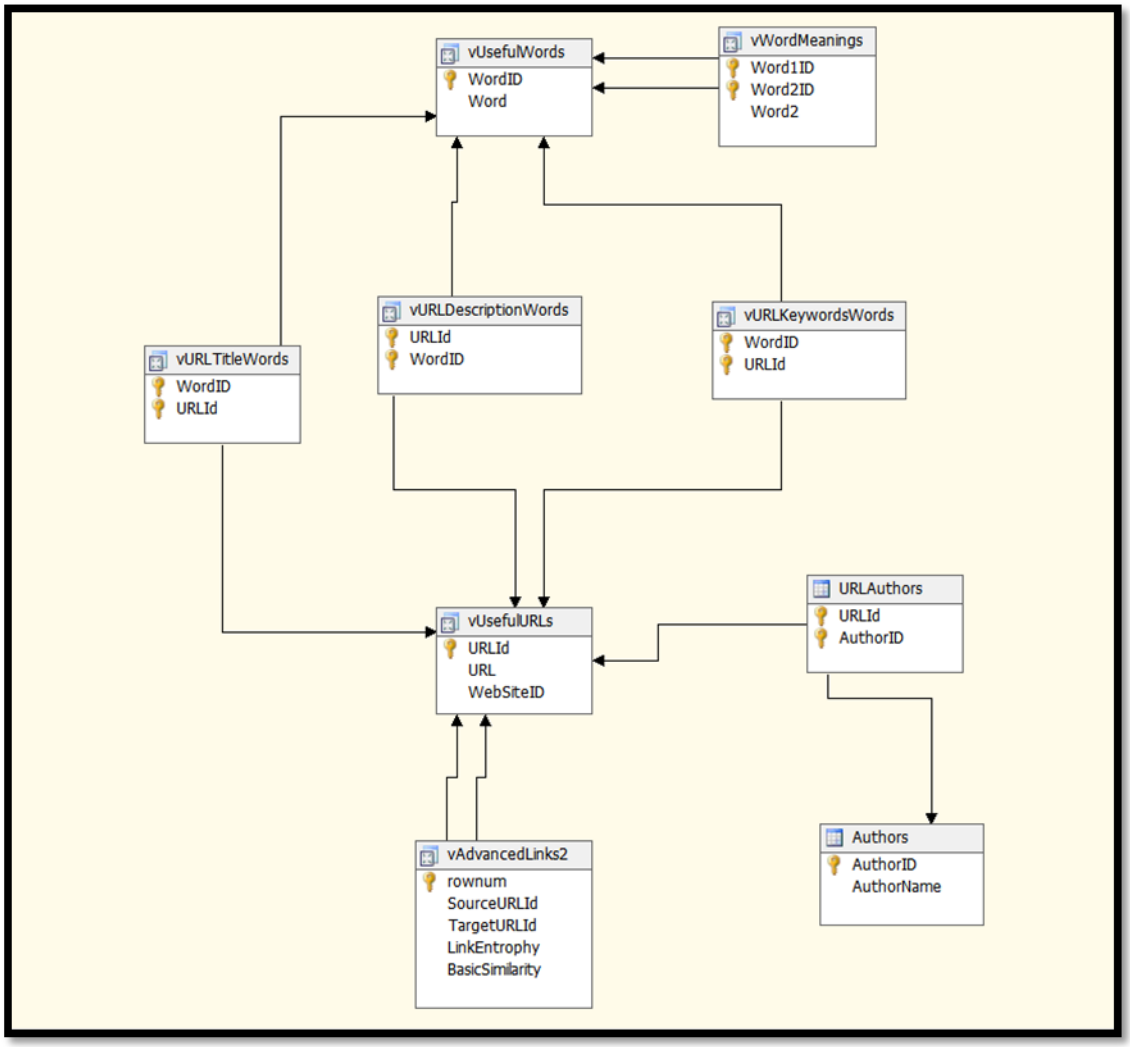
Above is the data mining process flow. Crawler processes are explained in previous sections. After crawler(s) finishes their jobs, an integration process must be applied. In this integration process, link levels are calculated.

In data mining model process, to a given model, data has been analyzed and categorized. To get prediction results, that model must be queried, depending on the query language on the selected data mining model. Desired result from query is saved on database as a table for further report generation.

Report generation contains post-data mining data alteration, calculations and listing them in a report.

2.5.1 Data Source and Views

To work with analysis services, data must be retrieved from a data source. Two separate thinks must be defined in project. A "DataSource" (DS) object is the connection between project and database server. "Data Source View" (DSV) object uses DataSource object to make a point of view to data source. After setting the connection, tables and views could be added to DSV. In DSV, virtual relationships and virtual keys could be created. For example, as in this project, vURLKeywordWords and vUsefulURLs are added to relationship on their URLId fields. This do not affect data base which as defined in DS.



Graphic 2.10 : Data source view

2.5.2 Mining Structure

In SQL Server 2008 Analysis Services, Data Mining Structure contains Data Mining Models. Those models can be Microsoft Clustering, Microsoft Decision Trees, Microsoft Naive Bayes etc. Mining structure is defined by a Case Table and its Nested Tables. Non-related tables in DSV could not be added to data mining structure. Each data mining model uses same structure of tables.

To train a model, data is split into two parts: Training and Testing. In this case it was unnecessary to use data for testing, because all of the data was used in training to get well defined cluster values.

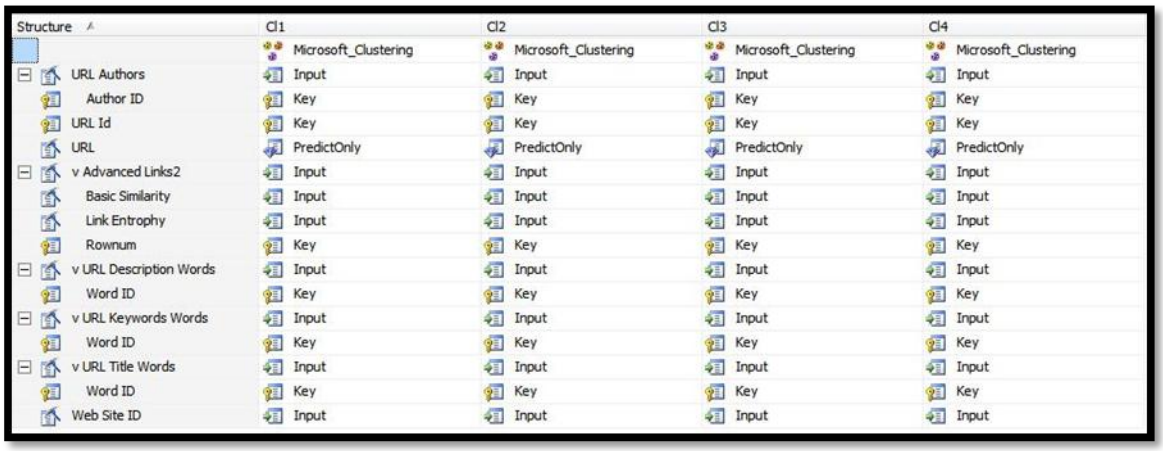
Data mining models could be parameterized for efficiency. Parameters are modified, to test which parameter values are best for this project. Four different parameterized models are created in mining structure for testing: Cl1, Cl2, Cl3, Cl4. Their names derived as the CLUSTERING_METHOD parameter values. Modified Clustering method parameters are listed in following table with parameter:

Table 2.1: Microsoft Clustering Algorithm Parameters

Parameter Name	**Description**	**Cl1**	**Cl2**	**Cl3**	**Cl4**
CLUSTER_COUNT	Specifies the approximate number of clusters to be built by the algorithm. If the approximate number of clusters cannot be built from the data, the algorithm builds as many clusters as possible. Setting the CLUSTER_COUNT parameter to 0 causes the algorithm to use heuristics to best determine the number of clusters to build. The default is 10.	0	0	0	0
CLUSTERING_METHOD	The clustering method the algorithm uses can be either: Scalable EM (1), Non-scalable EM (2), Scalable K-means (3), or Non-scalable K-means (4).	1	2	3	4

Details of EM and K-means and their scalability can be found on appendix.

Below is a graphic of how the columns are used in data mining model:



Structure	Cl1	Cl2	Cl3	Cl4
	Microsoft_Clustering	Microsoft_Clustering	Microsoft_Clustering	Microsoft_Clustering
URL Authors	Input	Input	Input	Input
Author ID	Key	Key	Key	Key
URL Id	Key	Key	Key	Key
URL	PredictOnly	PredictOnly	PredictOnly	PredictOnly
v Advanced Links2	Input	Input	Input	Input
Basic Similarity	Input	Input	Input	Input
Link Entrophy	Input	Input	Input	Input
Rownum	Key	Key	Key	Key
v URL Description Words	Input	Input	Input	Input
Word ID	Key	Key	Key	Key
v URL Keywords Words	Input	Input	Input	Input
Word ID	Key	Key	Key	Key
v URL Title Words	Input	Input	Input	Input
Word ID	Key	Key	Key	Key
Web Site ID	Input	Input	Input	Input

Graphic 2.11 : Data structure columns usage in mining model

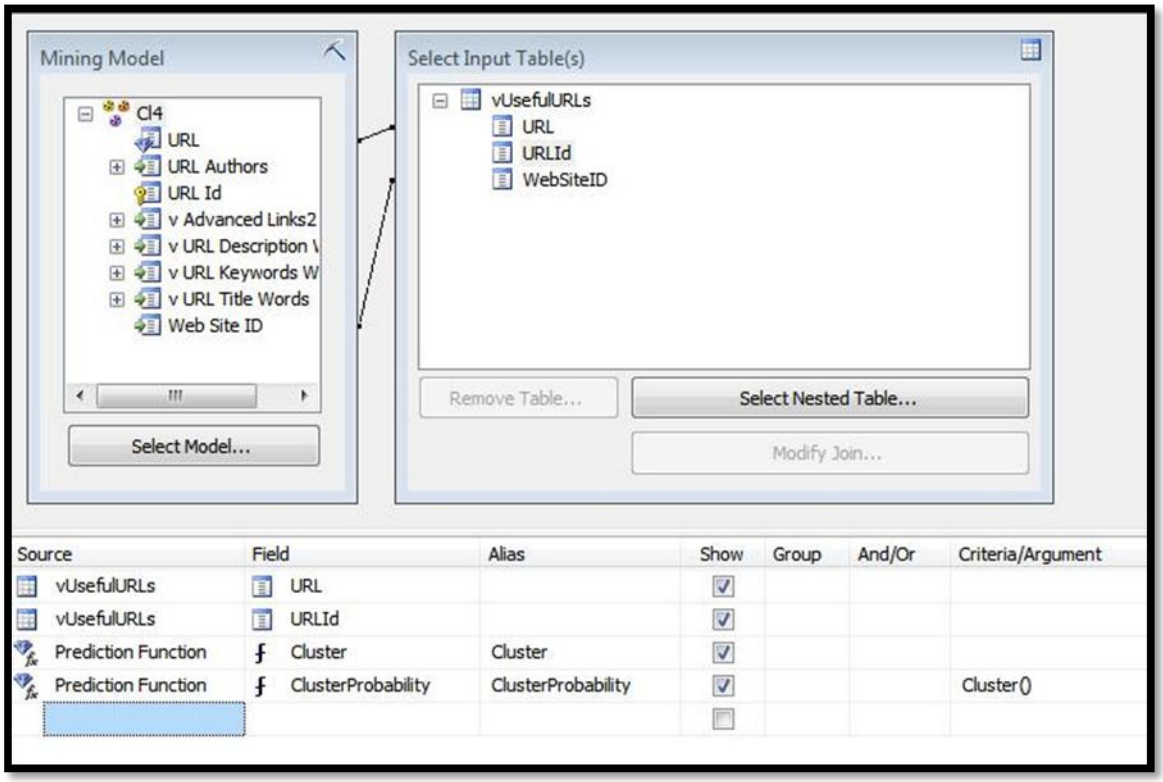
vUsefulURLs is the Case table. URLAuthors, vAdvancedLinks2, vURLDescriptionWords, vURLKeywordWords, vURLTitleWords are nested tables. For each nested table, a key column must be selected. Relating with case table columns of nested tables cannot be to model design. Thus SourceURLId and TargetURLId cannot be linked in model. An additional column required for model. Row number of links table used as Primary Key of Links table. This also ensured in DSV with virtual Primary key.

Every input column, while making model must be defined as which kind of data it is, such as continuous or discrete values. This is defined in Data Mining Structure. WebSiteID is defined as Discrete value; LinkEntrophy and BasicSimilarity are defined as Continuous values.

URL field of the vUsefulURLs is picked as predict column. Because URL field is the main column that results will be obtained. A column could be marked as predict and input column, however URL field is not selected as input, because URLId uniquely identifies the URL, which has picked as Key value of the model.

After setting structure, model and model parameters, model is ready to deployment and processing on SQL Server 2008 Analysis Services. Thus forms the clusters. Data

mining model can be queried with DMX language on SQL Server 2008 Analysis Services queries or Mining Model Prediction in Visual Studio 2008. Below are the query model, design and DMX code:



Graphic 2.12 : Data query model and design

```
SELECT
  t.[URL],
  t.[URLId],
  (Cluster()) as [Cluster],
  (ClusterProbability(Cluster())) as [ClusterProbability]
From
  [Cl4]
PREDICTION JOIN
  OPENQUERY([DSV],
    'SELECT
      [URL],
      [URLId],
      [WebSiteID]
    FROM
      [dbo].[vUsefulURLs]
    ') AS t
ON
  [Cl4].[URL] = t.[URL] AND
  [Cl4].[Web Site ID] = t.[WebSiteID]
```

Code 2.5 : DMX used in data mining model query

DMX query results in Visual Studio can be saved on SQL Server as a table. To determine, which Model is giving best results, each of them aggregated with Cluster Counts (CC) and Probability Average (PA). PA value is the most important value to determine which method is the best as long as amount of clusters are in scalable levels.

Table 2.2: Clustering Method PA & CC

Mining Model Name Used	Clustering Method	Number of Clusters (CC)	Average Probability of Clusters (PA)
Cl1	Scalable EM	6	0.865971614127324
Cl2	Non-scalable EM	2	0.961296156562331
Cl3	K-means	11	1
Cl4	Non-scalable K-means	11	1

K-means methods give better probability then EM. Scalability of K-means methods does not affect probability. From the given clue in Scalable EM and Non-scalable EM values, regular Non-scalable K-means method is favored.

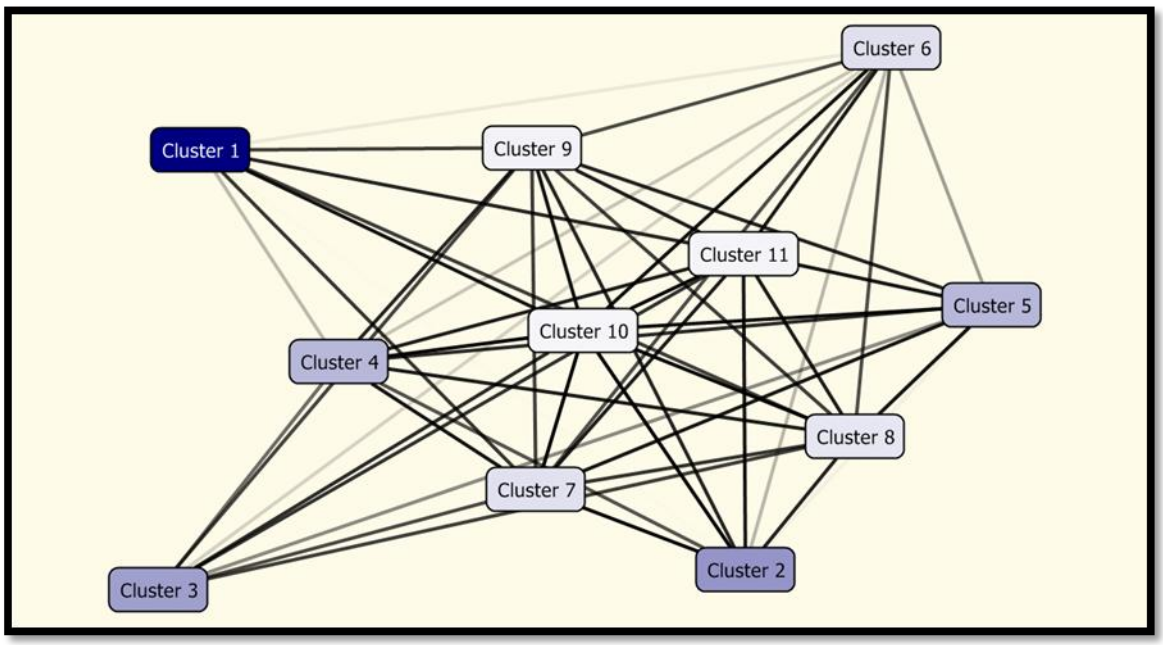
After deciding which model to use (Cl4), program is ready to get results.

3. FINDINGS & RESULTS

In this section, cluster diagrams, cluster result table examples and more derived findings will be displayed.

3.1 CLUSTERS

In the diagram bellow clusters in Cl4 are displayed. Darker clusters means, more population exists in it. Lines between clusters show how they are linked. Darker lines means stronger links.



Graphic 3.1 : Cl4 Cluster Diagram

3.2 QUERY RESULT

Below is the top 5 row result table from the query in previous section 2.5.2.

Table 3.1: Cl4 Results

URL	URLId	Cluster	ClusterProbability
http://www.battle.net/war3/nightelf/unitstats.shtml	1921	Cluster 3	1
http://www.battle.net/war3/races.shtml	1926	Cluster 3	1
http://www.battle.net/war3/ladder/	1928	Cluster 3	1
http://www.battle.net/war3/cheatcodes.shtml	1931	Cluster 3	1
http://www.battle.net/war3/nightelf/buildings.shtml	1935	Cluster 3	1

3.3 WORD STATISTICS OF CLUSTERS

Though URLId → WordID in URLKeywords, URLTitle & URLDescription, Cluster → Word matching can be done. Below is word – cluster table. In this table given WordProbability value is the probability of URL existing in the cluster. This is taken as the same of word probability. Shown values are distinct.

Table 3.2: Cl4 Words

Cluster	Word	WordProbability
Cluster 1	games	1
Cluster 1	warcraft	1
Cluster 1	world	1
Cluster 2	information	1
Cluster 2	war	1
Cluster 3	elves	1

3.4 FINDING SIMILAR PAGES

Program must display similar pages to the page that user has selected at the start of processes. The pages, those are in the cluster same as user selected page are relevant to user selected page.

To display similar pages, first user selected page's cluster must be found. The URLs and results below are from test set:

Table 3.3: Cl4 – User Selected Page Stats

URL	URLId	Cluster
http://www.battle.net/war3/nightelf/unitstats.shtml	1921	Cluster 3

Table 3.4: Cl4 – Top 10 results in Cluster 3

URL	URLId	Cluster
http://www.battle.net/war3/races.shtml	1926	Cluster 3
http://www.battle.net/war3/ladder/	1928	Cluster 3
http://www.battle.net/war3/cheatcodes.shtml	1931	Cluster 3
http://www.battle.net/war3/nightelf/buildings.shtml	1935	Cluster 3
http://www.battle.net/war3/nightelf/ancientofwonders.shtml	1936	Cluster 3
http://www.battle.net/war3/nightelf/buildingstats.shtml	1937	Cluster 3
http://www.battle.net/war3/nightelf/basics.shtml	1938	Cluster 3
http://www.battle.net/war3/nightelf/advanced.shtml	1939	Cluster 3
http://www.battle.net/war3/nightelf/combos.shtml	1940	Cluster 3
http://www.battle.net/war3/nightelf/killthene.shtml	1941	Cluster 3

4. DISCUSSION & CONCLUSIONS

Similar pages are clustered with considerations of metadata word meanings and link levels. In each cluster, similar pages are grouped. With finding the cluster of user selected page, gives the relative list of pages. Displaying URLs in same cluster satisfies the proposed approach.

This thesis contributes to web mining with adding dictionary synonyms and cross-linguistic meanings web mining clustering.

Links table has altered a bit to get better results by means of the way what the project proposed. This forced the hand of data mining structure and gave more significant clusters. Otherwise data mining model will fail with resulting all URLs in one single cluster. Altered table contains physical direct, indirect links, and symbolic links. Non physical links are considered as symbolic links. Links hold a link level and metadata similarity value of page. Those continuous values were vital while deciding the clusters during the data mining.

This approach also gives similar pages, but do not give a similarity percentage or sorts them by a similarity value. Because K-means method split all the data like black & white. However, a joint work of K-means and EM can calculate that similarity value. K-means method gives the absolute similar pages to start page. Distances of points to reference points can be calculated by considering center points of clusters in EM as reference points. Distance of start page to other page can be found by using those positions. For future works, this can be done by sorting pages by distance values.

Eleven thousands web pages had been crawled and indexed. Every search had its own subset of internet. After searching similar pages, index data had been cleared. Remained irrelevant data from previous search could pollute the next search data mining process.

This can result in irrelevant additional clusters in clustering. Also it can pull a useful point from the originating cluster to irrelevant cluster.

REFERENCES

BOOKS

Esposito, D., 2006. *Programming Microsoft ASP.net 2.0 Application: Advanced Topics*. Redmond, Washington: Microsoft Press

Esposito, D., 2006. *Programming Microsoft ASP.net 2.0: Core Reference*.Redmond, Washington: Microsoft Press

PERIODICALS

Berendt, B., Hotho A., Mladenic D., van Someren M., Spiliopoulou M., *Semantic Web Mining State of the art and future directions,* Web Semantics: Science, Services and Agents on the World Wide Web 4 (2006) 124–143

Bradley, P.S., Fayyad,U.M., Cory C.A., Revised October 1999, *Scaling EM(Expectation - Maximization) Clustering to Large Databases,* Technical Report MSR-TR-98-35, Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA 98052

Conesa,J., Storey,V.C. and Sugumaran,V., 2008. Improving web-query processing through semantic knowledge. *Data Knowl. Eng*. 66(1): 18-34

Fensel D., Patel-Schneider, P., *Layering the Semantic Web: Problems and Directions.* 2002 International Semantic Web Conference. Sardinia, Italy, June 2002.

He,X. and Jhala, P., 2008, Regularized query classification using search click information. *Pattern Recognition Journal*, Vol. 41, No. 7, pp. 2289-2297

Jin,H. and Chen, H., 2007, SemreX: Efficient search in a semantic overlay for literature retrieval. *Future Generation Computer Systems*, Elsevier B.V.

Lee,P.W., and Tsung,C.T., 2003. An Interactive agent-based system for concept-based web search. *Expert Syst. Appl.* 24(4): 365-373

Nikravesh,M., 2006. Enhancing the power of the internet using fuzzy logic-based web intelligence: Beyond the semantic web. *Capturing Intelligence*, Volume 1, Pages 441-464,

Perugini, S.,http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6VC8-4PFDPPJ-1&_user=10&_coverDate=03%2F31%2F2008&_alid=782582945&_rdoc=31&_fmt=full&_orig=search&_cdi=5948&_sort=d&_docanchor=&view=c&_ct=394&_version=1&_urlVersion=0&_userid=10&md5=821ffcc96fd512be814b5b9cf1a27d1b - cor1 Symbolic links in the Open Directory Project. *Information Processing and Management*, 44(2), 910-930.

OTHER

Dublin Core Metadata Element Set,Version1.1, http://www.dublincore.org/documents/dces/ [cited August 2008]

Dublin Core Metadata Initiative, 2008, http://www.dublincore.org/ [cited August 2008]

Microsoft, ASP.net Developer Center, http://msdn.microsoft.com/tr-tr/asp.net/default(en-us).aspx [cited August 2008]

Microsoft, SQL Server 2008 Books Online, http://msdn.microsoft.com/tr-tr/library/ms130214(en-us).aspx [cited August 2008]

Microsoft, Visual C# Developer Center, http://msdn.microsoft.com/tr-tr/vcsharp/default(en-us).aspx [cited August 2008]

Sabancı University & University of California at Berkeley, Turkish Lexical Database Project, http://www.hlst.sabanciuniv.edu/TL/ [cited August 2008]

UKOLN, University of Bath, 2002. DC-dot CGI parameters, http://www.ukoln.ac.uk/metadata/dcdot/help/cgi-params.html [cited August 2008]

UKOLN, University of Bath. DC-dot Dublin Core metadata editor, http://www.ukoln.ac.uk/cgi-bin/dcdot.pl [cited August 2008]

Wikipedia, Cluster analysis, 2008, http://en.wikipedia.org/wiki/Data_clustering [cited August 2008]

Wikipedia, Floyd–Warshall algorithm, 2008, http://en.wikipedia.org/wiki/Floyd-Warshall_algorithm [cited August 2008]

Wikipedia, OWL Web Ontology Language Guide, 2008, http://www.w3.org/TR/owl-guide/ [cited August 2008]

Wikipedia, RDF Schema, 2008, http://en.wikipedia.org/wiki/RDF_Schema [cited August 2008]

Wikipedia, RDF Vocabulary Description Language 1.0: RDF, 2008, Schema http://www.w3.org/TR/rdf-schema/ [cited August 2008]

Wikipedia, RDF/XML Syntax Specification, 2008, http://www.w3.org/TR/rdf-syntax-grammar/ [cited August 2008]

Wikipedia, Resource Description Framework, 2008, http://en.wikipedia.org/wiki/Resource_Description_Framework [cited August 2008]

Wikipedia, Semantic Web, 2008, http://en.wikipedia.org/wiki/Semantic_Web [cited August 2008]

Wikipedia, 2008, Shortest path problem, http://en.wikipedia.org/wiki/Shortest_path [cited August 2008]

Wikipedia, 2008, Web Ontology Language, http://en.wikipedia.org/wiki/Web_Ontology_Language [cited August 2008]

Wikipedia, 2008,Cluster analysis, http://en.wikipedia.org/wiki/Data_clustering [cited August 2008]

Wikipedia, 2008, Expectation-maximization algorithm, http://en.wikipedia.org/wiki/Expectation_maximisation [cited August 2008]

Wikipedia, 2008, K-means algorithm, http://en.wikipedia.org/wiki/K-means [cited August 2008]

APPENDIX

EM Clustering

MSDN Library 2008:

> *In EM clustering, the algorithm iteratively refines an initial cluster model to fit the data and determines the probability that a data point exists in a cluster. The algorithm ends the process when the probabilistic model fits the data. The function used to determine the fit is the log-likelihood of the data given the model.*
>
> *If empty clusters are generated during the process, or if the membership of one or more of the clusters falls below a given threshold, the clusters with low populations are reseeded at new points and the EM algorithm is rerun.*
>
> *The results of the EM clustering method are probabilistic. This means that every data point belongs to all clusters, but each assignment of a data point to a cluster has a different probability. Because the method allows for clusters to overlap, the sum of items in all the clusters may exceed the total items in the training set. In the mining model results, scores that indicate support are adjusted to account for this.*
>
> *The EM algorithm is the default algorithm used in Microsoft clustering models. This algorithm is used as the default because it offers multiple advantages in comparison to k-means clustering:*

- *Requires one database scan, at most.*
- *Will work despite limited memory (RAM).*
- *Has the ability to use a forward-only cursor.*
- *Outperforms sampling approaches.*

> *The Microsoft implementation provides two options: scalable and non-scalable EM. By default, in scalable EM, the first 50,000 records are used to seed the initial scan. If this is successful, the model uses this data only. If the model cannot be fit using 50,000 records, an additional 50,000 records are read. In non-scalable EM, the entire dataset is read regardless of its size. This method might create more accurate clusters, but the memory requirements can be significant. Because scalable EM operates on a local buffer, iterating through the data is much faster, and the algorithm makes much better use of the CPU memory cache than non-scalable EM. Moreover, scalable EM is three times faster than non-scalable EM, even if all the data can fit in main memory. In the majority of cases, the performance improvement does not lead to lower quality of the complete model.*

K-Means Clustering

MSDN Library 2008:

> *K-means clustering is a well-known method of assigning cluster membership by minimizing the differences among items in a cluster while maximizing the distance between clusters. The "means" in k-means refers to the centroid of the cluster, which is a data point that is chosen arbitrarily and then refined iteratively until it represents the true mean of all data points in the cluster. The "k" refers to an arbitrary number of points that are used to seed the clustering process. The k-means algorithm calculates the squared Euclidean distances between data records in a cluster and the vector that represents the cluster mean, and converges on a final set of k clusters when that sum reaches its minimum value.*
>
> *The k-means algorithm assigns each data point to exactly one cluster, and does not allow for uncertainty in membership. Membership in a cluster is expressed as a distance from the centroid.*
>
> *Typically, the k-means algorithm is used for creating clusters of continuous attributes, where calculating distance to a mean is straightforward. However, the Microsoft implementation adapts the k-means method to cluster discrete attributes, by using probabilities.*

Scalable Method

Basically scalable method of K-means and EM is for RAM purposes. It takes a subset of data while training. If training set is not enough to form clusters, another subset data is added to training set.

CURRICULUM VITAE

Name : Alper Özışık

Address : Menekşe Evleri A 14 D Blok No:10 D:10

Esenkent – Büyükçekmece / İstanbul

Birth Place and Year : İstanbul, 1982

Languages : Turkish, English, German

First Education : Nurettin Teksan İlkokulu / Kalamış – İstanbul, 1993

Middle Education : Üsküdar Anadolu Lisesi, 2000

University : Bahçeşehir University, 2005

Work Life : Bahçeşehir Üniversitesi DSİ Project 2006–2008

: BEKO / Beylikdüzü – İstanbul, internship, 2005

: Bahçeşehir Üniversitesi, staj, 2002

: ENKA / Beşiktaş – İstanbul, computer departments of various enterprises

: Öger Tur / Erdek, guide, 1997