UNL Based Semantic Search Engine and its Strategies with Interactive ANalyzer

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Shivangi
(801232025)

Under the supervision of:
Parteek Bhatia
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*UNL Based Semantic Search Engine and its Strategies with Interactive ANalyzer*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Parteek Bhatia* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Shivangi

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Parteek Bhatia
Assistant Professor
CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

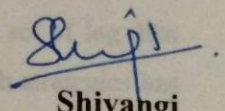
Acknowledgement

First of all I would like to express my special appreciation and thanks to my guide Dr. Parteek Bhatia, you have been a tremendous mentor for me. I would like to thank you for encouraging my research. Your advice on both research as well as on my career have been priceless. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. Sir taught me how to question thoughts and express ideas. His patience and support helped me overcome many crisis situations and finish this thesis.

Most importantly, none of this would have been possible without the love and patience of my family. My immediate family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my family. My extended family has aided and encouraged me throughout this endeavor. I have to give a special mention for the support and motivation given by my colleague and husband Mayur Vyas.

Dr. Deepak Garg is one of the best teachers that I have had in my life. He sets high standards for his students and he encourages and guides them to meet those standards.

I would like to acknowledge Vaibhav Agarwal for numerous discussions on related topics that helped me improve my knowledge in the area.



Shivangi

ABSTRACT

A semantic based approach is better in many ways than the keyword based searches. As the wide amount of information on the internet is exceling daily, the keyword based approach in many commercial search engines is failing to retrieve correct information without getting the knowledge and need of user query. Since the keyword based search could search for the particular language having that keyword, thus many commercial search engine is one language oriented.

The most challenging task is to design a system for multi-lingual machine translation environment, where large number of languages are to be translated between one-another. So, instead of following Statistical approach, an Interlingua based approach for MT is more preferable in multi-lingual machine translation environment. UNL is an intermediate language, thus works on the concept of relations between words having attribute associated with it. By gaining meaning of token information, it is possible to interpret a sentence rather than translating it to another language. IAN framework is used to analyze a query. This system gives user a freedom to check the results in more than one language with a single interface and at the same time. This helps in shortens the language barriers and help in searching cross-lingual.

We present our solution for performing cross lingual semantic based search. The project, that has been carried out in the thesis, present our semantic search engine of UNL, which encapsulate and sharpen previously proposed models. The use of UNL benefits us by providing feature of cross-lingual as well as multi-lingual search. We demonstrate that our model can predict search success more effectively than the existing state-of-the-art methods, on both our data and on a different set of log data collected from regular search engine sessions. For this different search strategies has been developed, which contributes in increasing recall of search results at the cost of decreasing precision, in manner of fall model. Together, our semantic search engine is another approach to perform not only a meaning based search but could also perform cross-lingual using IAN and EUGENE frameworks having dictionaries and rules of different languages.

TABLE OF CONTENT

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Introduction

The Internet is the largest repository of documents and information. To retrieve relevant documents to user query, searching is needed. With the use of search engine user can locate information quickly from a vast amount of knowledge available.

Unfortunately, many of the famous Search engines are monolingual or in English, the chief communication language worldwide. Although multilingualism is introduced in some of them but cross language searching is still not fully achieved. The hurdles in the path of achieving this objective were the language barrier and pattern matching approach of today's search engine.

The goal of any search is not the searching itself, the user want the precise and relevant information quickly and with as possible as minimum effort. Many traditional search engines follow summary approach while many other works on keyword based searching. This makes the engine not only with thousands of result which are generally not even seen beyond forty-fifty results but also makes the results language dependent. However many contextual based searching techniques are proposed but without following a proper and common approach for all languages, these techniques have narrow area. To remove that language barrier, it is hard to interpret or translate from each language to each another language without any intermediate language. With the steadily growing power and reliability of Natural Language Processing, the Universal Networking Language (UNL) can be a generous contributor in the realization of highly dependable search engines. This thesis aims to provide a semantic search engine using Universal Networking Language (UNL). Universal Networking Language (UNL) allows search engine to provide search results in many other languages too. Thus, it can perform cross-lingual search too. The first process analysis, enconvert whole document as well as the query into intermediate language, called Universal Networking Language (UNL). It uses IAN as UNLization which is better than UNLization(Enco), used in previous semantic search engine, as it could include D-rules contrast to Enco encoder. IAN is more users friendly. Various search strategies has been developed to understand the precision of document to the query entered by the user.

1.2 Motivation

As the information on Internet is growing and updated daily, it is becoming more and more difficult for the people to find the information they are looking for on the Internet, even with the help of search engines. It is because most of the current search engines do a pattern based search on the documents. They treat a web document and the query as nothing more than a bag of words. They do not try to find out the meaning of the query and pin-point the exact information the user is looking for. This results in retrieval of lots of irrelevant pages as well (problem of low relevance). And in some cases, many relevant pages are missed out (problem of low recall). Thus, a user has to search through hundreds of pages to and out relevant information.

Other than English, a sizable amount of information present on the Internet is in Indian languages such as, Punjabi, Hindi, Telugu, and Bengali etc. and in other Languages such as Spanish, French, and Russian etc. This creates a language barrier as the people who do not know these languages cannot access the information available in them. If a person can understand a language but couldn't write in it by its own, there is no way to find the query result in which the language define is the user mother language but the results are in different language or languages except to translate the sentence in mother tongue language to desirous languages. But it would become more hectic to convert each sentence in more than one language. World Wide Web is largest repository of knowledge known and a language gap here is obviously a big drawback.

In this report, with the help of Universal Networking Language (UNL), a cross lingual meaning based search engine is presented which can be used as a multi-lingual as well as cross-lingual platform for all sorts of search queries. Because the search engine is multi-lingual, it considers documents in different languages. The meaning based part of the search engine ensures that it gets all and only the relevant pages in response to his query whereas cross lingual part would search relevant pages in other languages.

1.3 Organization of Report

The solution for performing cross lingual semantic based search is presented in this thesis.

The next chapter, 2 introduces the Universal Networking Language (UNL), the

approach followed and the frameworks IAN and EUGENE which are used for the UNLization and NLization for the system. The focus is on some failed searches, which tend to frustrate many users of the commercial web search technology.

In chapter 3, the failure of some other meaning based search engines are discussed in brief.

Chapter 4 represents the problem statements.

Chapter 5 presents the semantic model of cross lingual semantic search engine, which encapsulate and sharpen previously proposed models.

Chapter 6 explains all the important strategies of searching to run the model perfectly.

Chapter 7 demonstrated that the proposed model can predict search success more effectively than the existing state-of-the-art methods, on both our data and on a different set of log data collected from regular search engine sessions.

The proposed semantic search engine is not only a meaning based search engine but could also perform cross-lingual search using IAN and EUGENE frameworks having dictionaries and rules of different languages.

Chapter Summary

Internet is a largest collection of the data. To get the required information from this data, the searching is required. For searching from that large collected data there is need of search engine. There are many search engines exists but problems with most of them are that either they are monolingual or based on English which is discussed in this chapter. When user gives the query to the search engine then his aim is not to find query itself, the user wants precise and relevant information according to his query. Many of search engines perform the keyword based searching and due to this type of searching, engine sometime finds thousands of results and out of them mostly are not desired. For resolving this problem there was a need of meaning based searching. The Universal Networking language (UNL) fits best for the meaning based searching, which is briefly discussed in this chapter.

Chapter 2

Introduction to UNL Framework

2.1 Introduction

The ideas and expressions in the mind of any human can be expressed by the communication, in which the medium, language, should be common. But if the users are using multiple languages then first thought comes into mind is the intermediate translator. Universal Networking Language (UNL) works as the intermediate translator. Universal Networking Language or UNL has been developed to serve the purpose of an intermediate language in the Interlingua approach adopted to overcome the language barrier. Universal Networking Language (UNL) is an electronic language for computers to express and exchange every kind of information. Since the advent of computers, researchers around the world have worked towards developing a system that would overcome language barriers. While many different systems have been developed by various organizations, each has its special representation of a given language. This results in incompatibilities between systems. It is therefore impossible to break language barriers all over the world, even if all the results are combined in one system. Against this backdrop, the concept of UNL as a common language for all computer systems emerged [1].

2.2 History

In 1995, at the Institute of Advanced Studies (IAS) at the United Nations University in Tokyo, three individuals had decided to establish the world's first virtual university. Within the setting of the IAS, a universal digital language had proposed, that is not derived from any natural language or any culture, but serve all of the conditions. These individuals had become the founding member of the UNDL foundation. And in this way, Tokyo became the centre to develop and create the UNL system. The specialized international scientists of linguistics and data processing have been called through symposium held around the world [2].

To protect the UNLizer and NLizer, the patent was filled in 1999. After getting the patent, the non-profit private-law foundation was decided to set in Geneva, Switzerland. With the decision of the owners of the rights to the invention and the inventors of UNL, Geneva becomes the center in 2001. Since then, the UNDL organizes schools and conferences to train the working behavior of UNL. Every year language professionals of many languages from different countries take part in schools and taking benefit of it. Now UNDL have a large amount of resource in different natural languages and UNL to continue the research and promote UNL [2].

2.3 The UNL workplace

The UNL workplace is the UNLweb. The researchers, developers, freelancers and language experts who have interest in NLP in general or UNL in particular can take part in creating, developing, editing and maintaining the UNL resources. Since the aim of UNL is to collect manpower of experts who can work and utilize the resources in common. Thus UNL invites them to share their expertise to create the UNL documents and project in different languages. For this except the training and university of UNL, virtual classes have been created. The UNLwiki is also available to learn more about the UNL and its working. Where working of different platforms have been discussed and explained. The working platforms include the UNLarium, the UNLdev, the UNLwiki and VALERIE (Virtual Learning Environment) which is introduced to train those wishing to be part of the project. UNLdev have the framework and tools such as IAN and EUGENE used in this project. VALERIE comprises two series of certificates; CLEA (CLEA250, CLEA450, CLEA700 and CLEA1000) and CUP (CUP500 and CUP1000). These certificates are required in order to be an active user and participate in the development of UNL projects as UGO-A1, NADIA-A1, and BRONO-A1 etc. These projects get open after the completion of the previous one in that natural language. Some of the projects are paid while for other the experts work as volunteer. No previous knowledge in UNL or NLP is required in order to create an account in the UNLweb environment. The UNLweb is entirely free and all its results are released under an Attribution Share Alike (CC-BY-SA) Creative Commons license [3]

2.4 Representation of language using UNL

Since proper dictionary is being created and each word should have a meaning and its attributes to be searched in text retrieval, thus it is better say, it can use as interpretation of the sentences. Sentence by sentence the information is represented in UNL. Then in semantic hyper-graph that information is represented. Each node represents the concept having some semantic attribute. The nodes having semantic relation, work as arc in hyper-graph, are written according to the syntax of UNL having elements.

UNL can be said to comprise of the following elements:

- Universal Words (UW)
- Attribute
- Binary Relations

The structure of UNL is defined by the UNL specification. It specifies the structure of a UNL document; the syntax of Universal Words; the set of attributes; the set of relations; and all the information concerning UNL as formalism [4].

The descriptions of these are given below.

2.4.1 Universal Words (UW)

Universal Words, abbreviated to "UWs", are character strings representing simple or compound concepts. They are the nodes of a UNL hyper-graph. A UW is not only a unit of the UNL syntactically and semantically for expressing a concept but also a basic element for constructing a UNL expression of a sentence or a compound concept. Such a Universal Word is represented as a node in the semantic network of UNL expression [5].

They are labels for concepts, syntactic and semantic units to form UNL Expression. A combination of a set of UWs - linked with each other through relations and modified by attributes - expresses the meaning of a sentence [6]. Format of a universal word of UNL is given in (2.1).

<uw> =:: <headword> [<constraint list>] ……. (2.1)

Here, a headword of a UW is an English expression, a word, a compound word, a phase or a sentence of English. If the meaning of a headword is unique, the headword itself becomes a UW. Otherwise, constraints are attached to the headword to make more specific UWs. If a UW consists of a headword only, it is called a "Basic UW" [7].

2.4.2 Attributes

Attributes are mainly for the purpose to describe subjectivity information. It includes time, aspect, emphasis, focus, topic, attitude, feeling and judgment. Attributes are also used to specify qualities of concepts such as the genericity, the specificity and the logicality of UWs. They are attached to a UW or a scope to specify the information [8]. Types of attributes are shown in Table 2.1.

Table 2.1: Types of Attributes of UNL

Describing logicality	@transitive,@symmetric,@identifiable,@disjointed
Describing times	@future,@past,@present
Describing aspects	@begin,@complete,@continue,@end,@progress,@state,…
Describing genericity and specificity	@generic,@def,@indef,@not,@ordinal
Describing emphasis, focus and topic	@emphasis,@entry,@focus,@topic,…
Describing attitudes	@affirmative,@imperative,@interrogative,@request,…
Describing feelings and judgments	@ability,@grant,@wish,@will,@obligation,@possible,@regret,…

Example

"Health problems and their resolution"

UNL expression of above example is given in (2.2).

{unl}

and(resolution(icl>solution):0S.@entry, problem(icl>abstract thing):09.@pl)

mod(problem(icl>abstract thing):09.@pl, health(icl>state):02)

mod(resolution(icl>solution):0S.@entry,they(icl>thing):0M.@pl)
{/unl} ...(2.2)

In the UNL Expression given in (2.2), has *'and'* and 'mod' are relations. 'resolution(icl>solution)', 'problem(icl>abstract thing)', 'health(icl>state)' and 'they(icl>thing)' are UWs. '@entry' and '@pl' are attributes. A scope node is used as a way to refer to a scope [8]. In our example, there is no need of taking any scope. A graph of UNL expression corresponding to (2.2) is shown in Figure 2.1.

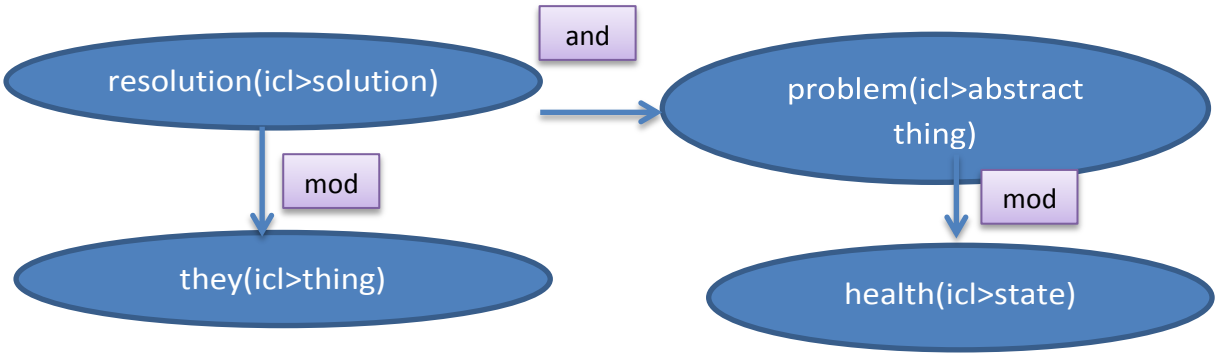


Figure 2.1 UNL graph of expression given in (2.2)

2.4.3 Relations

Binary relations are the building blocks for UNL documents. The semantic relations between the UWs form the arcs in the UNL hyper-graph. The relations between the UWs have different labels according to the different roles they play. In UNL, relations have been normally used to represent semantic cases or thematic roles between UWs [8].

There are 47 relations in the UNL, such as '*agt', 'gol', 'obj*', *etc*. They are used to connect every two UWs or scopes to construct the semantic networks of UNL expression. The relations are edges in the UNL graphs, or functions of the directed binary relations that constitute UNL expression [8]. A relation denotes a semantic role of a UW or scope for others. Relations in UNL are given in Table 2.2.

Table 2.2: Relations of UNL

agt	and	aoj	bas	ben	cag	cao	Cnt	cob	con
coo	dur	Euq	fmt	frm	gol	icl	Ins	int	iof
man	met	mod	nam	obj	opl	or	Per	plc	plf
plt	pof	pos	ptn	pur	qua	rsn	Scn	seq	shd

src	tim	tmf	tmt	to	via				

2.5 Tools for UNL Interpretation

UNL tools are programming software that is developed to assist linguists in producing UNL resources (such as dictionaries and grammars). These tools are not tailored to non-specialists and require expertise in UNL [9].

IAN and EUGENE are the frameworks for analyzing and generation of a language to and from UNL respectively. The process of converting a source natural language into UNL expressions is called UNLization, whereas, converting the UNL expressions into a target language is called NLization.

Two files are used for the process of UNLization and NLization. The first file is a dictionary file that lists correspondence between the Universal Words (UWs) of UNL and the words of a native language (natural language). The second file lists grammatical rules, combination of T-rules and D-rules. Each of these files is specific to a particular language and is developed under UNLdev [10].

2.5.1 IAN (Interactive ANalyzer)

IAN is a natural language analysis system. It is used to represents natural language sentences as semantic networks in the UNL format. In its current release (version 1.1), it is a web application developed in Java and available at the UNLdev [10]. It includes the natural language analysis grammar and operates semi-automatically with the help of dictionary in natural language and grammar rules. Language specialist do the word sense disambiguation, but the system filters the candidate using the disambiguation rules' optional set in grammar called D-rules. Natural language analysis grammar is responsible for syntactic processing automatically, although syntactic ambiguities are signaled to the user if any, which can backtrack and select different syntactic paths. In all the cases, it is optional to do human interaction, and thus used to improve the results. If the case is of no human intervention, the system consider the grammar and in the lexicon [11]. IAN performs the three following movements over the input file.

- Segmentation, i.e., the division of the input document into a series of processing units (sentences), which are processed one at a time.
- Tokenization, i.e., the identification of the tokens (lexical items) of each sentence of the input document.
- Transformation, i.e., the application of the transformation rules of the grammar over each tokenized sentence in order to represent it as a UNL graph.

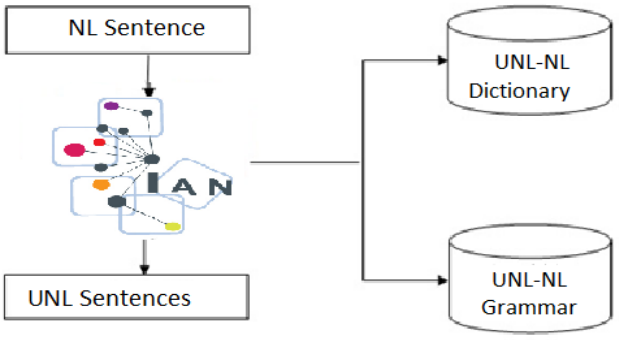


Figure 2.2 Processing of NL sentence

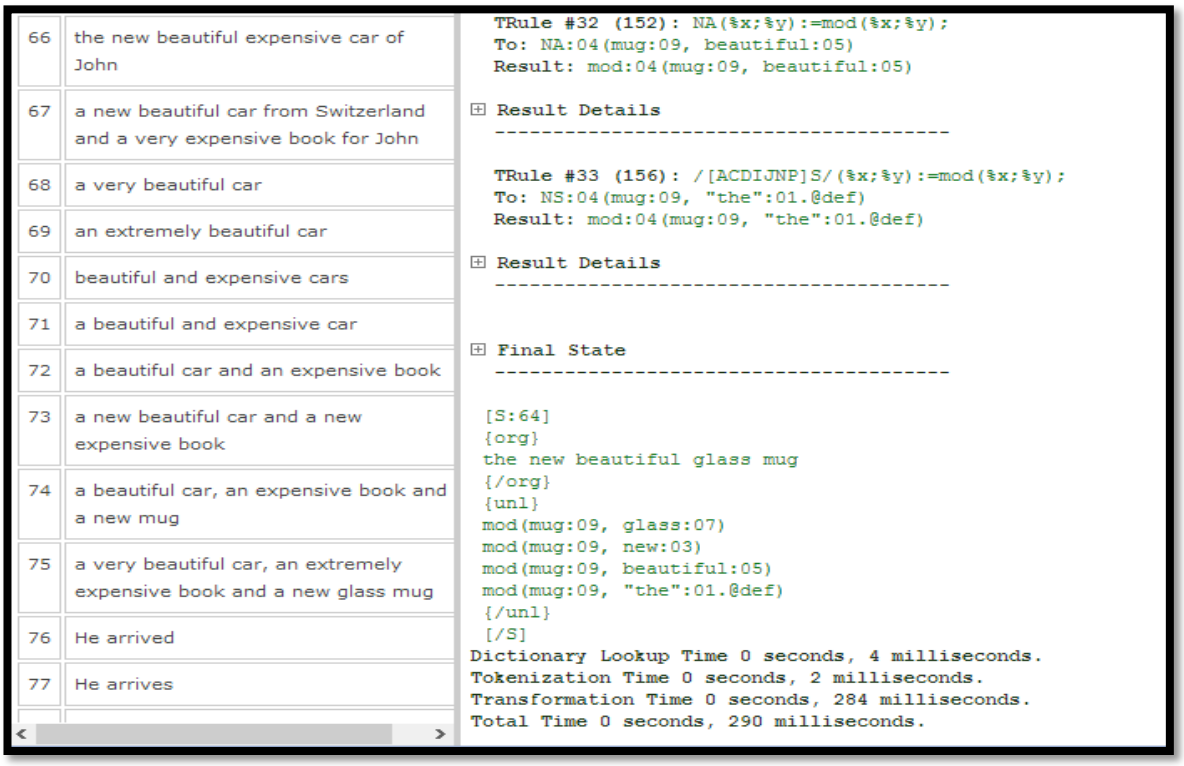


Figure.2.3 IAN Framework

Basically IAN UNLize a natural language to UNL with the help of Rules and dictionary.

NL – UNL Dictionary: For each natural language a separate dictionary is used. Dictionary includes the lemmas and its information such as part of speech, gender, singularity etc.

NL – UNL Grammar: For each natural language, there are separate grammar rules called T-rules. There could be ambiguity rules too, called D-rules, which may or may not be language specific.

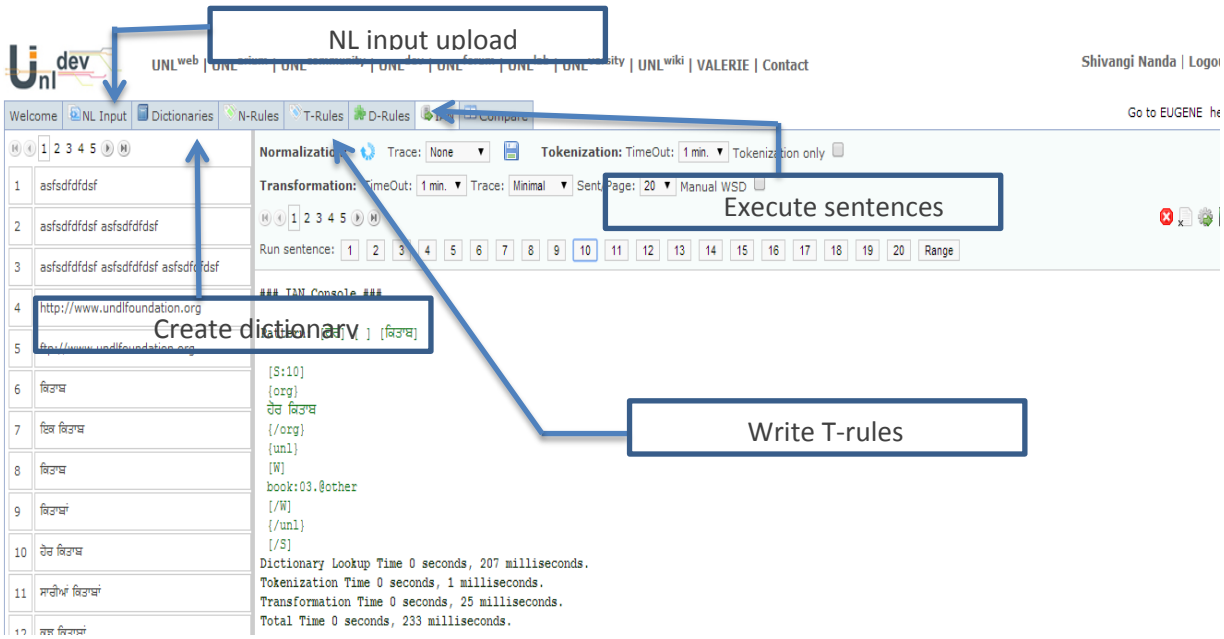


Figure.2.4. Snapshot of showing tabs of IAN framework.

After Login into the UNL website [10][28], IAN framework is available at UNLdev. The following tabs are visible now:

1. NL input: The natural language file can be either created manually or one can upload it. The natural language file should contain sentence by sentence in new line. One can see the previously uploaded or created file in this tab. Left hand side the sentences can be viewed with sentence number, in IAN execution tab.

2. Dictionary: One can create all the dictionary words of natural language coming in the NL input file. The different files can be selected once a time to increase availability of the words. It is important to provide dictionary entry of each word in

NL file else by default the word will behave as temporary word, lead to unwanted behavior in execution.

3. T-rule: T-rules are important to decide how bunches of words would behave and interpreted to UNL.

4. Execute IAN: all the NL-input file's sentences execute here. One can select the number of sentence to run or the number of sentences to run. These would behave according to the T-rules created earlier. Each sentence could be converted into UNL sentences.

2.5.2 EUGUENE (dEep-to-sUrface GENErator)

EUGENE is a natural language generation system that generates natural language sentences out of semantic networks which is represented in the UNL format. In its current release (version 1.1), it is a web application developed in Java and available at the UNLdev [10]. EUGENE is a fully automatic natural language generator that takes an UNL input and provides the output without any intervention of human in natural language. Similar to the UNLization tools, it also is a language-independent and should be parameterized to the natural language input through a grammar and a dictionary, provided as separate interpretable files [11]. Similarly, EUGENE performs the three following movements over the input file.

- Segmentation, i.e., the division of the input document into a series of isolated graphs, which are processed one at a time.
- Tokenization, i.e., the identification of the tokens (UWs, relations and attributes) of each graph of the input document.
- Transformation, i.e., the application of the transformation rules of the grammar over each tokenized graph in order to generate a natural language sentence.

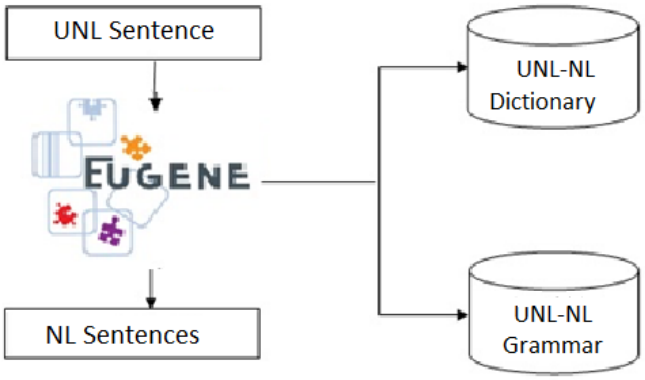


Figure 2.5 Processing of UNL sentence

2.6 Importance of UNL, as intermediate language

To understand language translation, let us consider the Vauquois triangle, shown in figure 2.6. For the same word pattern languages the effort of translation is less compared to syntactically different languages. So, the direct translation works only for the languages having same word pattern. For difference in syntax or semantics, more steps are needed to translate a language from one to another. Deciding and translating need more time. Thus intermediate translation is the solution of all.

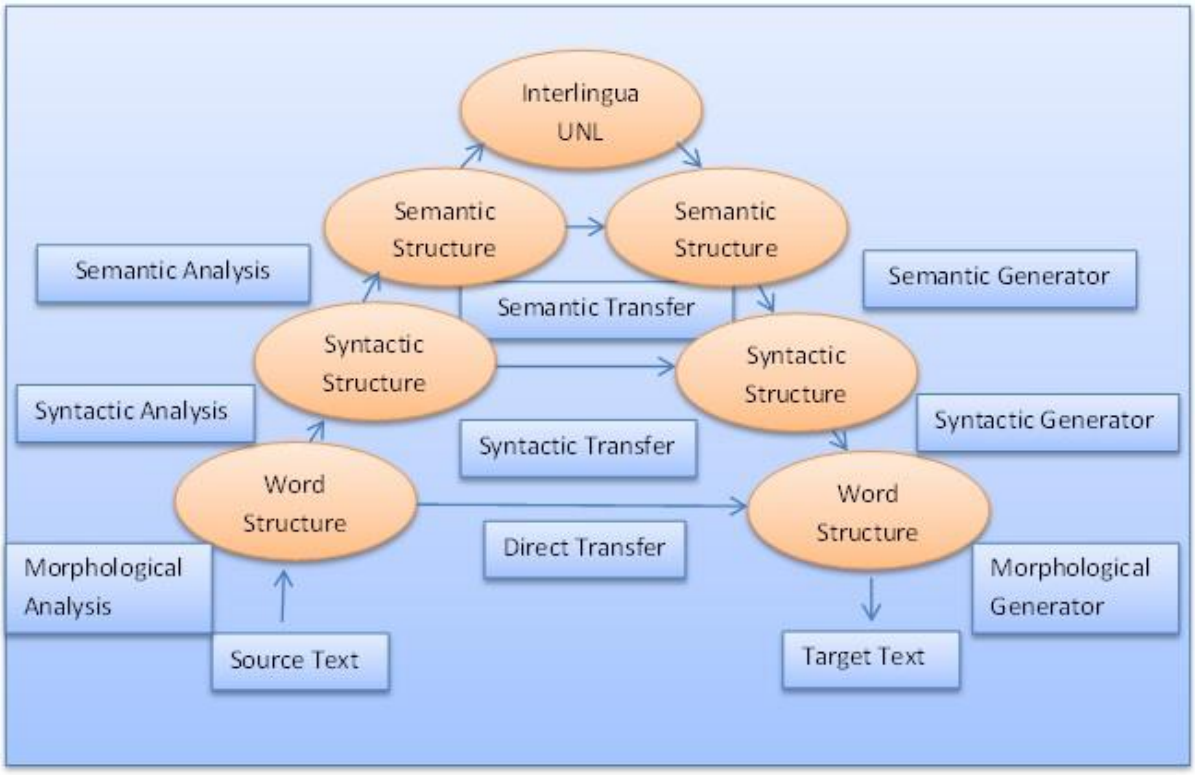


Figure 2.6 Vauquois triangle

It is clear from the figure 2.6 that if UNL worked as the intermediate language there would be no need of checking the word structures, semantic structure or syntactic structures of the language. But we must have grammars and T-rules involved with the source and destination languages with respect to the UNL language, each for analysis as well as generation that is responsible for all this.

The UNL as the intermediate language helps in 2 steps translation. As shown in Figure 2.6, the first step is of analysis from the source natural language to the intermediate language, UNL. And the next step is generation of target natural language sentences from UNL. The tools available are:

- EnConvertor and IAN
- DeConvertor and EUGENE.

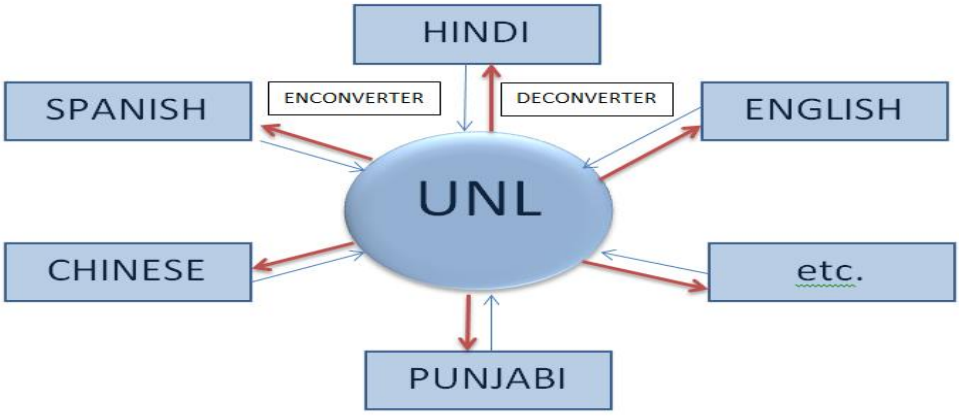


Figure 2.6 An UNL System

Chapter summery

In this chapter a brief introduction about the Universal Networking Language (UNL) is given. UNL is created in the 1995 at the Institute of Advance Studies at the United Nations University in Tokyo. UNL is a digital language that is not derived from the any natural language but it serves all the conditions of it. In 1999 a patent form is filed to protect the UNLizer and NLizer. After getting the patent the UNDL foundation decided to set in Geneva, Switzerland in 2001. Now UNDL have a large amount of resources in different natural languages and UNL. In this chapter UNL workplace is represented where researchers, developers, freelancers and language experts can take part in creating, developing, editing and maintain the UNL resource. There are also some terminology are used in this chapter about the UNL like UNLwiki, UNLarium, UNLdev, VALERIE etc. UNLwiki is used to learn about the

UNL, UNLarium and UNLdev provides the working platform for the UNL. VALERIE stands for Virtual Learning Environment which is used to train those wishing to be the part of the project. Also some terminologies like Universal Word (UW), attribute and binary relations are discussed briefly in this chapter. In this chapter, UNL interpretation tools IAN and EUGUENE are also discussed in briefly.

Chapter 3

Literature Review

This chapter represents the famous existing search engines related to thesis. This allows putting our model in perspective of the field.

3.1 Existing Search Engines

Google is widely believed to be the best traditional search engine, although it searches for some common languages but there is not any concept of cross lingual searching since it's a text based search engine that takes a query and documents as nothing more than a store of meaningless patterns. Also, a lot of useless and sometimes garbage information is returned, which not only take up a lot of computing and bandwidth resources, but also is very cumbersome. It uses the famous Page Rank algorithm [12], which ranks the documents according to the hyperlink structure, coupled with the local query specific score to give the final rank to a page [13].

3.2. Existing Meaning based Search Engine

There are very few meaning based search engine. Though the attempts are good but there results are not close to the mark and hence failed to achieve a commercial success. A few of such attempts are search engines like oingo.com, simli.com and excite.com [14], said to do meaning based searching.

3.2.1. Oingo

Oingo, Launched in October, 1999, has already introduced three fully functional products Domain Sense, Direct Search and AdSense. Domain Sense is an Oingo's meaning-based domain name suggestion technology. It currently increases domain name sales for leading registrars around the world. Direct Search, a meaning-based search technology, uses the company's ontology to provide more precise and effective search results. AdSense serves the most highly targeted advertisements on the Internet; effectively targeting advertisements based on search meanings rather than keywords. But none of these search engines is a true meaning based search engine as none of them considers the meaning emerging from the interconnection of words.

Their results for many queries are sometimes even bad than Google, which, as we know, is not a meaning based search engine.

3.2.2 HAKIA

Hakia[15] semantic search is essentially built around three evolving technologies:

- OntoSem (sense repository)
- QDEX (Query indexing technique)
- SemanticRank algorithm

OntoSem is Hakia's repository of concept relations. The QDEX is Hakia's replacement for the inverted index that most engines use to save web content. QDEX extracts all possible queries relating to the content and these become the gateways to the original document. This process greatly reduces the data set that the indexer has to deal with while querying data on-the-fly. Finally, the SemanticRank algorithm independently ranks content on the basis of more sentence analysis. Credibility and age of the content is also used to determine relevancy. The engine has also started using the Yahoo BOSS service and also presents results in a "gallery" with categories for different content matching the query [15].

Figure 3.1 Interface of HAKIA semantic search engine

3.2.3. Sensebot

Sensebot [16] prepares the text summary according to the user's search query. It identifies key semantic concepts by using text mining algorithms that parse the Web Pages. The retrieved multiple documents are then used to perform a coherent summary. This coherent summary becomes the final result for user's query. The main sources for these results are usually the news agencies.

Figure 3.2 Interface of Sensebot semantic search engine

SenseBot represents a search engine which does not provides the links or Web pages as output of your query, instead of that, it gives the summary of the result, which is desired, from the web pages where your query matches. Generally SenseBot is good for the queries where user wants brief information about any topic or wants to understand something; in that case SenseBot gives the desired summary of this topic which helps to understand it.

3.3. Existing Multilingual Search and Information Retrieval resources

There are 1000s of languages all around the world. Meaning based searching is not totally depends upon token based text searching, thus gives a scope of providing better multilingual searching. There is various approaches lead to that goal. Some of them are briefly described below.

3.3.1. MIRTH

A Multilingual Information Retrieval Tool Hierarchy (MIRTH) [17], gave a general model of multilingual information retrieval for Web searching. It is coped with English and Chinese information retrieval. It first created an index file that contained key information about different Webpages and then indexed both document titles and document contents. Users could give queries of search terms directly via a Web browser. Then, the tool started a search program that explored the pre-computed index files in real time and then yielded search results accordingly.

3.3.2. MULINEX

Multilingual Web Search and Navigation Tool [18], is developed by a consortium consisting of five European companies. MULINEX supports English, French, and German. It also supports selective information access, browsing and navigation in a multilingual environment. The project emphasizes on a user-friendly interface, which not only supports the user by presenting search results but also thematic category, information about language, automatically generated summaries, and also allows the user to sort results by multiple criteria.

3.3.3. Kazhugu

A multilingual Internet search engine, claimed to be India's first multilingual search engine in regional languages.It has been developed by Anna University-KB Chandrasekhar (AU-KBC) Research Centre, Chennai. 'Kazhugu' means eagle in Tamil. This search engine is ready for use in Tamil websites. The Research Centre will soon come out with similar Internet search engines for Hindi, Malayalam, Telegu and Kannada languages. The search engine is placed for testing on the Internet portal of Sify Ltd [19].

3.3.4. MIETTA

MIETTA, Multilingual Information Extraction for Tourism and Travel Assistance, is a project whose objective is to develop a cross-lingual information management platform. The system resulting from the MIETTA project will allow retrieval of tourist information in several languages (English, Finnish, French, German, Italian) and on a number of different geographical regions (the German federal state of Saarland, the Southwestern Finnish region centered around Turku and the Italian city of Rome) [20].

3.4. UNL based search engine.

3.4.1 AgroExplorer

It claimed as a first UNL, meaning based searched engine for Indian rural languages. Its basic application is for agriculture. It performs query answer system especially for tackling queries of farmer regarding agriculture. Its EnCoder(ENCO) and DeCoder(DECO) tools are not powerful to tackle many application. Its application is also restricted to agricultural area [21]. Although the searching strategy is common and base to all UNL based search engines.

3.4.1.1 Searching Strategy

To provide the relevant data, the two most important measures precise and recall is used. It uses complete searching, which is precise and most relevant document searcher, partial searching, used to find the documents having some of the concepts matched with the user query, used to increase recall value. Before these techniques comes into picture, it is important to understand query graph and sentence graph, used in these searching.

Query Graph and Sentence Graph

In a text based search engine, if the query XY (X and Y are words) is given, a page containing both the terms X and Y will be more relevant than a page having only X or only Y, assuming the Page Rank is same for both pages.

In UNL based semantic search engine both, the document and the query, is converted into UNL with IAN framework. With the help of Indexer XML file, query graph is generated assuming first universal word as X and second universal word as Y and there is a relation between these two, which binds it into a single form.

First of all, find the query graph. After the query graph is found, a sub graph checking is done on every sentence graph in the document. If the query graph is sub graph of a sentence, that sentence is considered relevant to the query. The document getting high number of sentence matches comes up first in the result list.

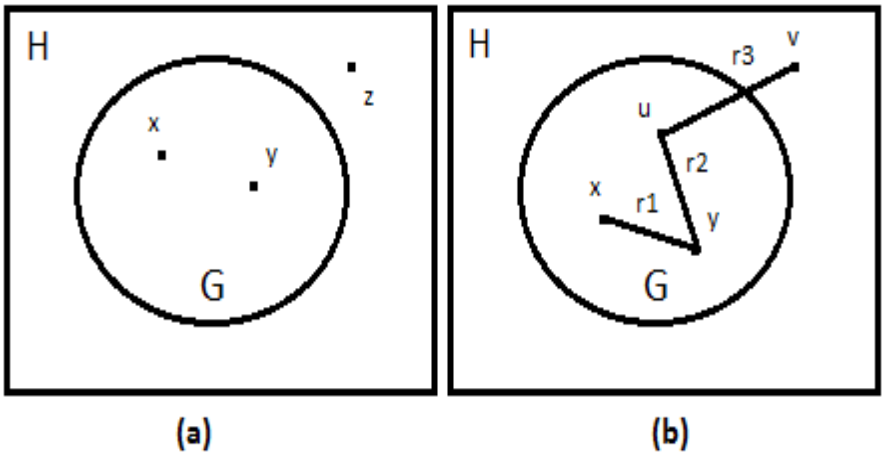


Figure 3.3 (a) Showing Sentence Graph for text based searching (b) Showing Sentence graph for meaning based searching.

In the figures above x, y, z, u, v represents the tokens in case of text based search and UWs in case of meaning based search. It is clear from the figures above that meaning based search resolves the relation between UWs and have r1, r2 and r3 to show relation between them.

Here H shows the boundary of a document and G is the sentence graph.

3.4.1.2 Complete Matching Search

Complete matching signifies that the whole meaning of query is in the sentence of searched document i.e. the precise document comes up. The condition signifies that searched document must have all the information of the searched query. In technical way we can say that complete match searches for all the relations of query graph to be in sentence graph hence making all the UWs in the sentence graph. Thus, essentially, we have a collection of UNL graphs of document and a given UNL graph of query, which needs to match with the document. If the query UNL graph is a sub graph of any sentence UNL graph in the document, then we can say that the document containing that sentence is completely relevant to the query and the search engine should retrieve it. Thus complete matching confirms the total precise calling of the relevant document. Mathematically, it can be shown as,

$$Rq(d) = \frac{\sum_{s \epsilon Sd} r(s)}{Sd}. \quad (1)$$

Where, Rq(d) =relevance of documented to the query q.

Sd =the set of sentences in the document d.

r(s) =relevance of sentences to the query q.

If the query is matched to the sentence then rq(s) =1 else 0, i.e. a sentence will be either relevant to a query or it will not be. It will lead it to high precision but low recall. It also matches the exact relation in the universal words. To make it more flexible, similar relation match technique is used.

To get the result of the query, we have to search all the sentences of all the documents. It is not only the time consuming process but lead to an impractical way as the documents are in very large number over internet. Thus indexing is required to find the results for a given query.

To make it indexed we divide the whole corpus according to relations and their respective universal words and maintain an XML file of document number and sentence number of the relation and their universal word.

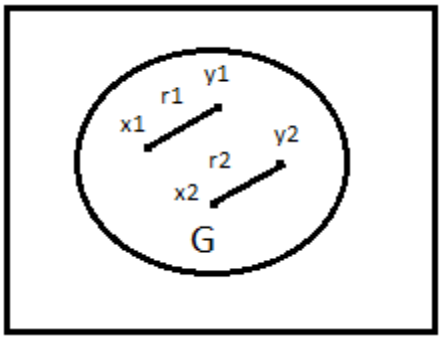


Figure. 3.4 Complete search graph

In figure 3.4, x1, y1 and x2, y2 are UWs having the relationship of r1 and r2 respectively. For a sentence to be said complete matching search of a query, all the UWs and relations in a query, should be in that sentence.

3.4.1.3 Partial Matching Search

Complete search ensures one-or-none matching. This result into only relevant sentences is searched by search engine. But this could be a drawback in case of recall. To overcome this problem, partial matching scheme is being introduced which has lower precision but high recall value.

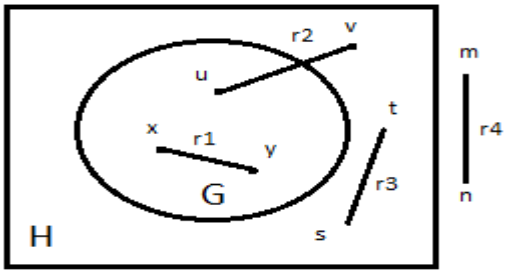


Figure 3.5 Partial search graph.

Partial search is beneficiary if UNL of query have more than one relation. In figure 3.5, H shows the boundary of the corpus document and G represent Sentence graph. x and y are UWs having r1 relation in query. In the same way, u and v have r2 relation, s and t have r3 relation and m and n have r4 relation. Any sentence in document comes in partial search if any one relation and its UWs, fully matched with one or more than one query relations and its UWs.

According to this phase, the result comes with, if document have sentence matched with a part of the query. Equation for finding relevance of document remains same as in above approaches. Relevance of sentence can be defined as,

$$r(s) = \alpha \frac{n}{N} + (1 - \alpha)\frac{l}{L}. \quad (2)$$

Where,
r(s) =relevance of sentences to the query q.
α =empirical constant.
n= number of relation edges found, of the query, in the sentence.
N =Total number of relation edges in the query.
l= number of correct links in the sentence.
L =Total number of links between all UWs in the query.

3.4.2 UNL Explorer

It has been developed under UNDLfoundation[23]. It also uses encoder (ENCO) and decoder (DECO) tools. Sometimes it provides the unwanted search results. It provides multilingual results which are converted from a single language to another single language. It has multilingual dictionaries in 47 languages and work on UNL ontologies to perform search [22].

Chapter Summery

In this chapter literature review, represent some existing work done in the field of search engine. There are some famous token based search engines like Google and meaning based search engines like Oingo, Hakia, Sensebot are represented. Google is one of the most popular search engine, although it does not performs the cross lingual searching. It is a token based search engine; it does not understand the meaning of the query given by the user. Since it does not understand the meaning of the query, sometimes it returns the garbage information. There are also some meaning based search engine exists like Oingo, Hakia and Sensebot which does the meaning based searching to some extends but they are not truly a meaning based search engine as discussed in this chapter. There are also some existing multilingual search and information retrival resources like MIRATH, MULINEX, Kazhugu, MIETTA, AgroExplorer and UNL Explorer are discussed in this chapter.

Chapter 4

Problem Statement

Although it is easy to search just tokens, but to reduce the irrelevant matching, semantic search is needed. Existing problems in the other systems and previous exist semantic search engine has been discussed below.

4.1 Issues with traditional search engines

The safe and easiest way of searching is searching the tokens in the query. This type of search engine doesn't know what the paragraph is all about, but collect the keyword from it. When the token in the query matches the keywords in the paragraph, the page is retrieved according to the relevancy and ranking scheme of the search engine. The basic lagging of this type of search engine has been discussed below.

4.1.1 Based on Keywords or Token

Many current search engines, including Google, perform searching on keywords. A query is treated as list of keywords and any entries containing any or all of those words are returned.

For example:

Query: "handwritten map of treasure"

Expected: A map to find a treasure which is hand written

Output: On Google [24]

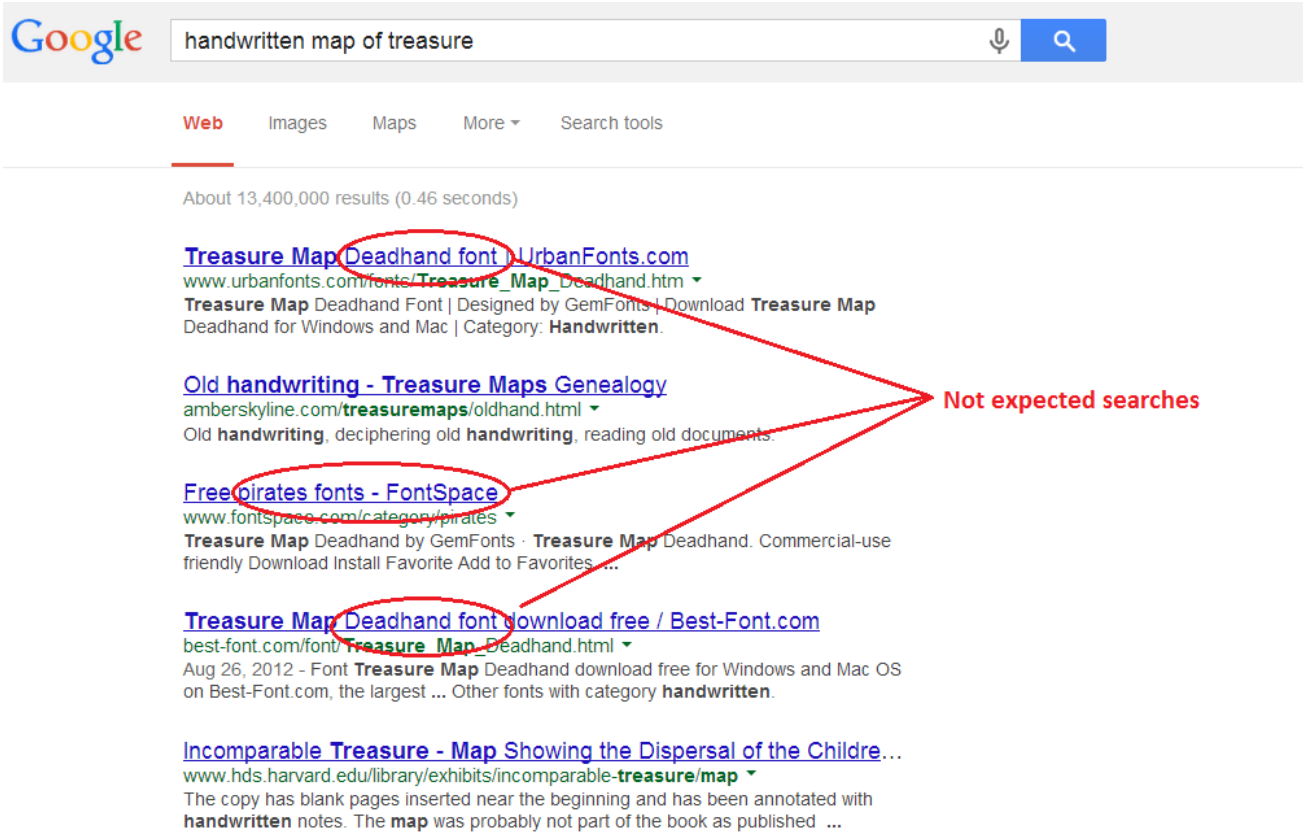


Figure 4.1 Unexpected search results in Google first page [24]

4.1.2 Limited to use for a single language search

Since the keyword based searches considered tokens as their search input. That makes it limited to search the query word to the same natural language documents in which that query word belongs to.

4.2 Issues with other semantic search engine.

Researchers developed many other semantic search engines. But either the techniques are so restricted that the projects are not successful or the search engine is restricted to only a specific domain.

4.3 Solutions of the problem

Problem stated above needs a strong meaning based search engine. UNL could be the best approach to perform this. Since UNL is an intermediate language and language independent, thus can provide not only meaning of query and sentences but also the multi-lingual and cross-lingual capabilities.

4.3.1 Meaning Based Search

UNL uses meaning representations for example, ‘date’ could be a time or a fruit. But

in UNL, relations and attributes are there to differentiate between meanings. It is easy to differentiate the meaning of not only the homonym words but is easy to get the meaning of sentence or query.

4.3.2 Cross-lingual search

UNL is an intermediate language thus it is easy to convert from a language to another. Also it interprets a sentence rather than direct translation of it thus meaning of sentence remain same. UNL solves many ambiguities comes in machine translation.

Consider a situation of creating a cross-lingual system without the use of UNL. It would be required to translate from a language to another. From each language to another 2 components is needed for translation, one for analysis and another for generation. Thus for n number of languages, n^2 components would be needed. But with the use of UNL, an intermediate language, we need just two components for each language. Thus it makes the search in cross-lingual approach with 2n components.

Chapter 5

Objective and Methodology

While concerning the use of multiple languages in searching, which facilitate the user to search in any of its known natural language, the result may cumbersome thus, results into high complexity. We discussed earlier how the use of universal networking language (UNL) reduces the complexity of involving multiple languages. It is the best option to perform the tasks related to interpretation. To include the feature of cross-lingual approach, UNL is the better option over other. Cross-lingual system is different from multi-lingual in the sense that in cross-lingual approach, user can give query in any natural language recognized by the system. And the system will print the result in all the language selected by the user rather than a single language from multiple language choice, as in multilingual approach.

5.1 Objectives

The main objective of research work is to create an UNL based search engine which perform semantic based search by using Interactive Analyzer (IAN). In order to perform this task, following objectives were proposed to be carried out.

i) To study the working of previous UNL based search engines.

ii) To develop a powerful Indexer and preprocessor.

iii) To develop an XML creator which creates the XML of the whole corpus and helps in maintaining the UNL index in .doc form.

iv) To develop a searching module which overcomes the issues with previous UNL based search engine.

v) To create NL-UNL Dictionary for Corpus EOLSS and corpus UGO-A1 using IAN framework.

5.2 Methodology

To achieve the objectives discussed in section 5.2, IAN framework developed by UNDL Foundation has been used.

i) To achieve the first objective, various UNL based search engines have been searched and explored. For this purpose, the reference of AgroExplorer[21],

claimed first search engine, and UNL explorer[22], latest UNL based search engine, have been taken.

ii) A powerful Indexer with the pre-processor has been developed. This is capable of making Index file of large UNL corpus. To check its working two corpuses EOLSS, with 25000 sentences of natural language English(en), and UGO-A1 with 300 sentences in 8 different language namely, Afrikaans(af), Baatonum(bba), Bulgarian(bg), English(en), Estonian(et), French(fr), Hindi(hi), Punjabi(pa), have been considered. The indexer successfully creates an XML index file having record of all the relations, first and second universal words having that relation, and indexing schemes.

iii) XML creator has been developed. This is responsible for creating XML file of the index with index elements. XML creator works on both side *i.e.* it creates XML file of the UNL of the query as well as the XML file of the UNL of pre-processed documents. Finally, the two XML files would match according to the search techniques used and the results showed to the interface of the user.

iv) In previous UNL based search engine, AgroExplorer[21], there is only two searching schemes. The complete matching, finds the exact semantic matching of sentences in document and query. But this result in very few documents comes under search. To improve recall, partial matching search has developed. Partial matching search gives some reliability over precision.

But these searching techniques sometimes fail to get the documents having the data related to the query. The reason being the relation used by EnConverter in query is different from the relation used in the sentences of the document, although they convey same meaning. For this purpose 'Similar relation matching' searching has been developed which gives another chance to Complete matching. A list of similar relations has been maintained and swapping of relations has applied to confirm the complete matching covers most of the document which should come under search.

Similarly, Universal word matching searching techniques have been developed which is different from keyword based matching in a sense that different keyword may have common universal word.

vi) To create NL-UNL Dictionary for Corpus EOLSS and corpus UGO-A1 using IAN framework, all the words coming under EOLSS and UGO-A1 has been entered according to the dictionary entry rule of IAN framework. The dictionary entry of UW "box" as shown in (5.1) has been explained in below.

[ਬਕਸੇ]{*-1*}"box"(LEX=N, POS=NOU, NUM=SNG) <eng, 0, 0>; …5.1

Here, "box" is the UW, 'ਬਕਸੇ' bakse is its meaning in Punjabi. Its lexical category is noun, number information is given to be singular. In IAN all the forms of 'box' in Punjabi would be stored to make a common box as its universal word(UW).

After writing dictionary entries, T-rules have to write. Write T-rule or Transformation rules based on the characteristics of the UWs included in the UNL sentence.

Chapter 6

Building Block of the Search Engine and Implementation

6.1 Introduction

This chapter describes the overall structure, data and control flow of the cross lingual meaning based search engine. To simplify and easy to use, the system is divided in various modules. Each module is created separately and in the end, joined with an interactive user friendly interface. The project is created in JSP to make it more functional and could be deployed as live interactive search engine by introducing better web crawler. Since the aim of the project is just to show the benefits of semantic search engine using UNL and to show how it can overcome the problems of other search engine, it is being tested on corpuses. These corpuses are available at UNLarium [25]. Various modules that make up the search engine and brief description of their implementation are presented here.

6.2 Corpus

The World Wide Web, having over 1000 million pages, continues to grow at a million pages per day. A large amount of text changes every month. This has posed serious problems of scale for the crawler and the search engine.

Undoubtedly, we need a general purpose crawler for a general search engine, but presently we are considering only the corpus given under UNLarium [25] as the search engine's domain. This is because we require some more research and time before IAN can convert all of the English sentences into UNL successfully. So, till that is done, we decided to focus on a narrow domain for our search engine.

A giant, general purpose web crawler is neither necessary nor sufficient for this purpose. It will lead to a lot of time and space wastage. Thus a corpus could be sufficient. For the purpose of problem, the Encyclopaedia of Life Support Systems (EOLSS), the corpus from the project UNL-EOLSS under UNLarium [25], consist of 30 articles of the Encyclopaedia of Water, one of the many encyclopaedias of EOLSS [26]. To check multilinguality, UGO-A1 corpus is referred which is in 15 different languages and is being developed under UNLarium [25]. The corpus attached in Appendix A and B.

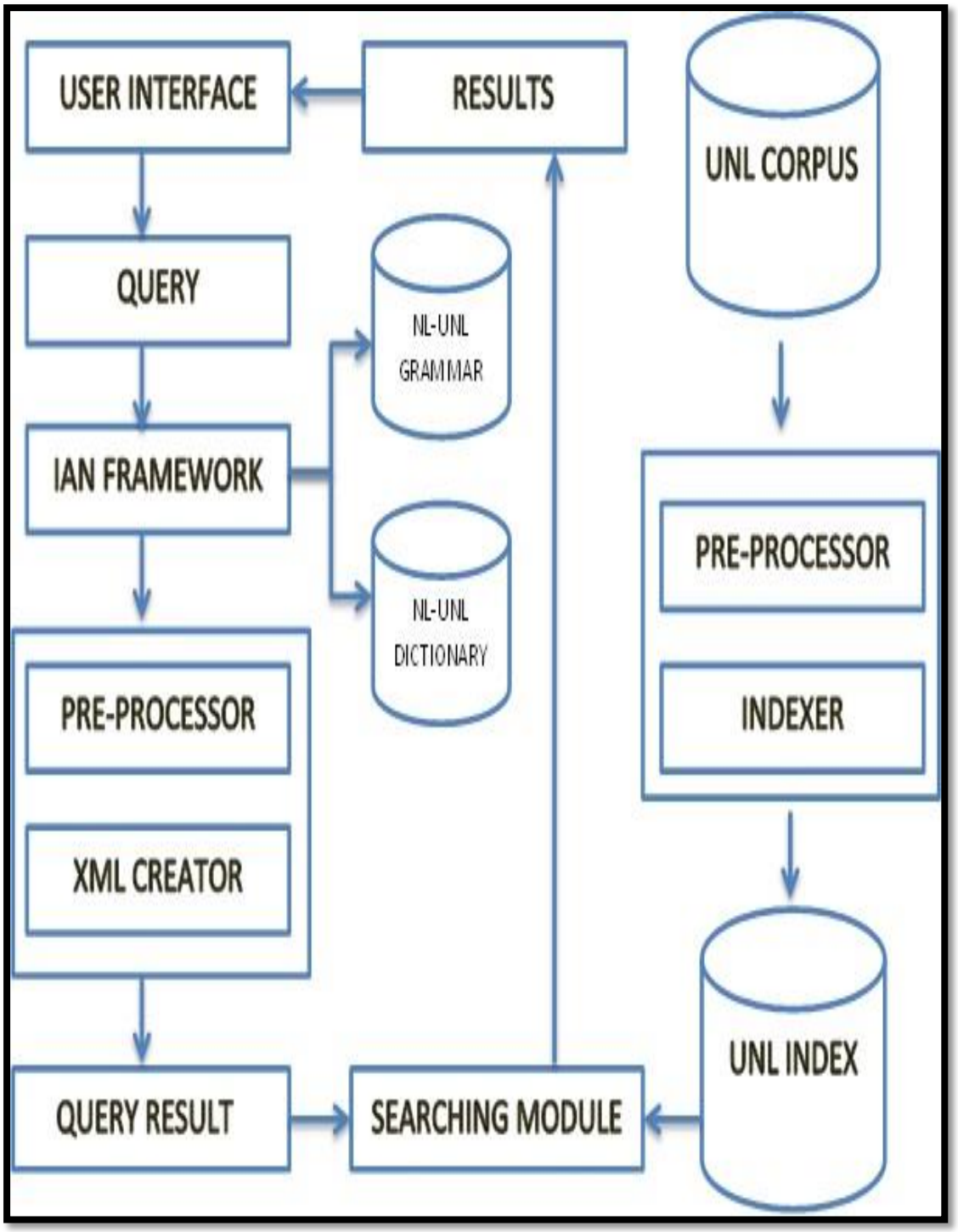


Figure 6.1 Block Diagram of Search Engine.

6.3 IAN Framework

For the purpose of UNLization, IAN Framework is being used [27]. It takes the query from the user in any predefined natural language, convert it in UNL and pass it to the pre-processor module of search engine. Since now, the search engine with IAN framework is semi-automated thus manually the UNL is passed to the search engine.

6.4 EUGENE Framework

This module is similar to the Enconverter module except that it will convert the UNL representation back to any natural language .It will be required to convert the documents which the search engine returns in the user's language to show cross-lingual by maintaining its meaning. Since the project is semi- automated, the work of EUGENE framework is being stored in corpus.

6.5 Preprocessor

The main task before indexing any file is to preprocess it according to the need and process. It not only fulfills the first objective but also useful in creating index files easily. This module preprocesses the UNL documents and converts them to an intermediate form which is then indexed by the indexer. The UW-ID is assigned to all the universal words as shown in Figure.6.2. It helps in making the document and query in specific similar structure. This, again, in turn calls for an indexer for indexing the documents in the XML file. The pre-processor module is used for processing both, the UNL of query and UNL expressions of corpus.

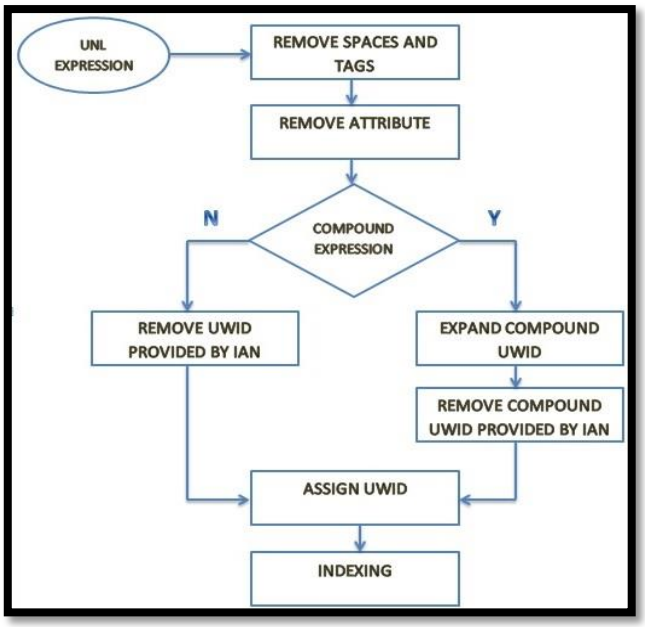


Figure 6.2 Flow diagram of Pre-processor Module

Following steps are performed by this module.

1. All unnecessary spaces and tabs are removed.

2. All attributes of the Universal Words (UWs) are removed. Removing the attributes do not affect the search results and in some case may even improve the recall of the search engine.

3. Some of the UWs are not assigned any UW-ID by the Enconverter software and there is probability of having different UW-IDs for the same UW, thus UW-IDs can be omitted from the UNL expressions when a UW is unique in a UNL expression. For consistency in indexing procedure, a unique dummy UW-ID is assigned to all the UWs. This also simplifies the partial matching algorithm.

4. All the compound UW ids are replaced by the actual UNL expression of all the relations in the sub graph. All the normal UW ids present inside such a sub graph is removed before substituting it for the compound UW id.

5. The UW id of such a sub Graph is kept as the compound UW id representing that sub graph in the original UNL expression.

6. All the compound UW ids are removed. For example, the sentence is “an extremely beautiful car”. Its input UNL expression is shown in figure 6.3.

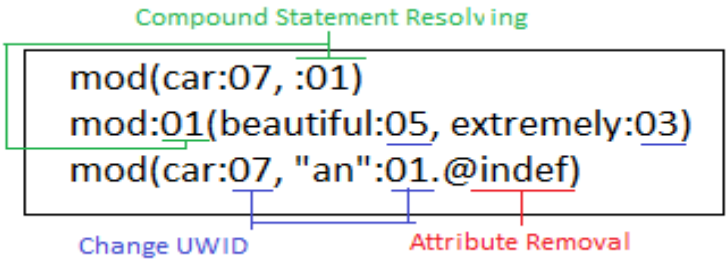


Figure 6.3 UNL expression

After performing preprocessing steps discussed above, the resulting output after these steps will be shown in figure 6.4.

Figure 6.4 Pre-processing output of UNL expression

6.6 Indexer Module

Indexer module is responsible for generating an UNL index, which is used by the search module to quickly find the relevant documents and calculate their relevance.

Basically, indexer separates the resulting UNL expressions got from the preprocessor into relation, first UW, second UW, first UWs id, second UWs id. It than updates the UNL index with these values, which is again a XML file. To provide efficient data to the query of user, the meaning representation *i.e.* UNL expression is needed. The UNL documents are indexed in UNL index. It is kept in a XML file named *index_unl*. The each UNL document is indexed and information is stored in UNL index. The fields of *index_unl* are shown in table 6.1.

Table 6.1 UNL Document Index

Fields	Description
Docid	UNL document id
Language	Language of original document
Numlines	Number of sentence in the document

The UNL index stores the actual index of UNL expressions. Each entry keeps the information of UNL relation, its UW1 and UW2 and their respective UW-IDs and the sentence number and its document id where the combination of relation, UW1 and UW2 occurs. The UNL index file is shown in Table 6.2

Table 6.2 UNL Index

Fields	Description
rel	Relation name
UW1	First parameter of the relation.
UW2	Second parameter of the relation.
UW1ID	ID associated with UW1.
UW2ID	ID associated with UW2.
sent	Sentence number.
docid	UNL document id in which above fields occur.

The following example illustrates the index representation of the UNL expression of one sample sentence.

Docid: EOLSS12

sentence:

[S:1174]

{org:en}

A potentially advantageous method of storing water underground in times of water surplus to meet demand in times of shortage.

{/org}

UNL expression:

{unl}

tim(method(icl>way):0T.@entry.@indef, time(icl>period):1W.@pl)

tim(demand(icl>need):2R, time(icl>period):31.@pl)

man(time(icl>period):1W.@pl, underground(icl>how):1H)

man(advantageous(aoj>thing):0G, potentially:04)

mod(surplus(icl>amount):2B, water(icl>liquid):25)

aoj(advantageous(aoj>thing):0G, method(icl>way):0T.@entry.@indef)

pur(surplus(icl>amount):2B, meet(icl>satisfy(agt>thing,obj>thing)):2M)

mod(method(icl>way):0T.@entry.@indef, water(icl>liquid):1B)

mod(time(icl>period):31.@pl, shortage(icl>situation):3A)

mod(time(icl>period):1W.@pl, surplus(icl>amount):2B)

obj(store(icl>accumulate(agt>thing,obj>thing)):13.@progress, water(icl>liquid):1B)

obj(meet(icl>satisfy(agt>thing,obj>thing)):2M, demand(icl>need):2R){/unl}

[/S]

The UNL index is shown in Figure 6.5. Each entry (UNL expression (rel, uw1, uw2)) points to the pair of document id and its sentence number where it occurs.

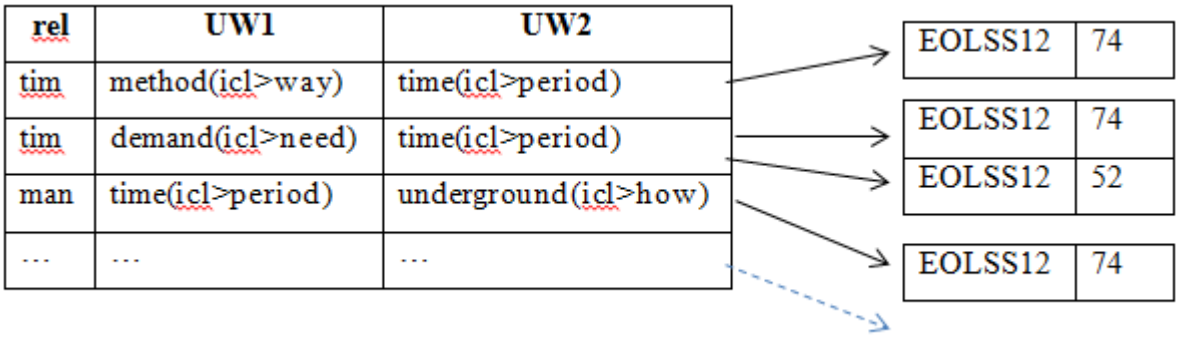


Figure 6.5 Index of UNL expressions.

6.7 Search Module

This is the main module and the major objective of the search engine. It takes as input the UNL index generated by the IAN module and a query (in UNL form) and returns a list of documents matched according to decreasing order of their relevance along with lines matched. This module uses the same preprocessor used by the indexer module to preprocess the UNL query. Below the search strategies have been discussed.

Fall model of Search Strategies

The more precise data should come up in the list of document searched. Since there may be a possibility of thousands or millions of search results come out, when a query is fired. So it is important to sort the result in decreasing order of their relevancy. Since search strategies present here often shows the same kind of behavior, the searching performed in a sequence to get precise data at top. For e.g. we get the most precise results with the complete query search, so we performed complete query results first. Since similar relation match gives the same benefit of complete search results by providing another chance to query results, thus similar relation match comes after complete query match. In the same way partial query match comes after the similar query match and so on. The fall back model is shown in Figure 6.6

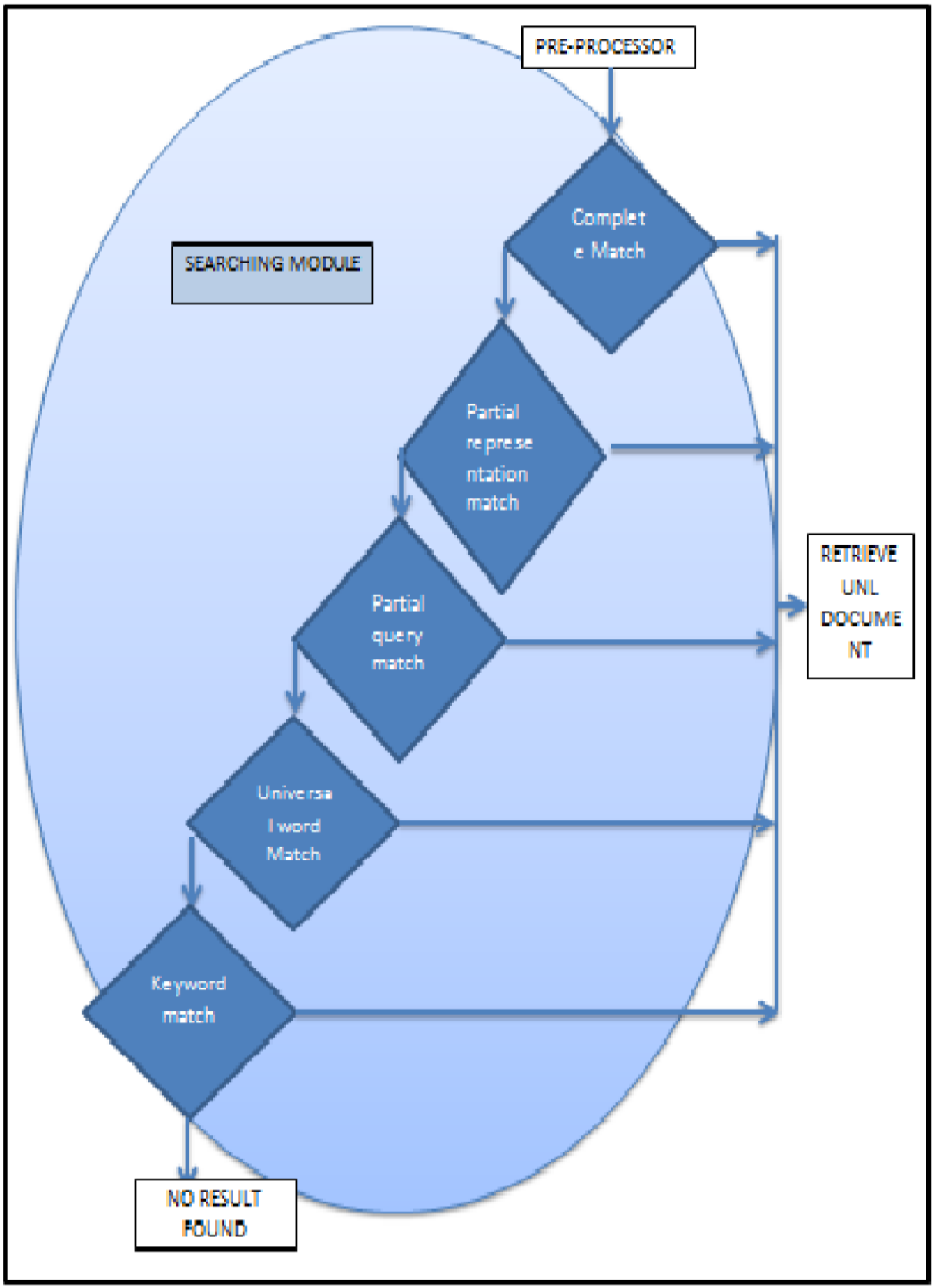


Figure 6.6 Fall model of search strategies.

6.7.1 Complete Matching

The simplest and more precise search result searching is through complete matching. As discussed in section 3.4.1.2, complete matching search the most precise and relevant sentences only. So the sentence is either accepted by the complete search result or fails. Let us take an example, "the beautiful car"

Query: "the beautiful car".

UNL: "mod(car.@def, beautiful)".

Explanation: By pre-processing the query by the preprocessor used by the indexer. The resulting UNL of query is:

mod(car:0A, beautiful:0B)

Here, 'car' and 'beautiful' are the UWs, bind by a relation 'mod'. When we perform Complete search, the sentences in the corpus, having all UWs(car and beautiful in this case) and relation(mod in our case), matched with query UWs and relation respectively, should come up when searching.

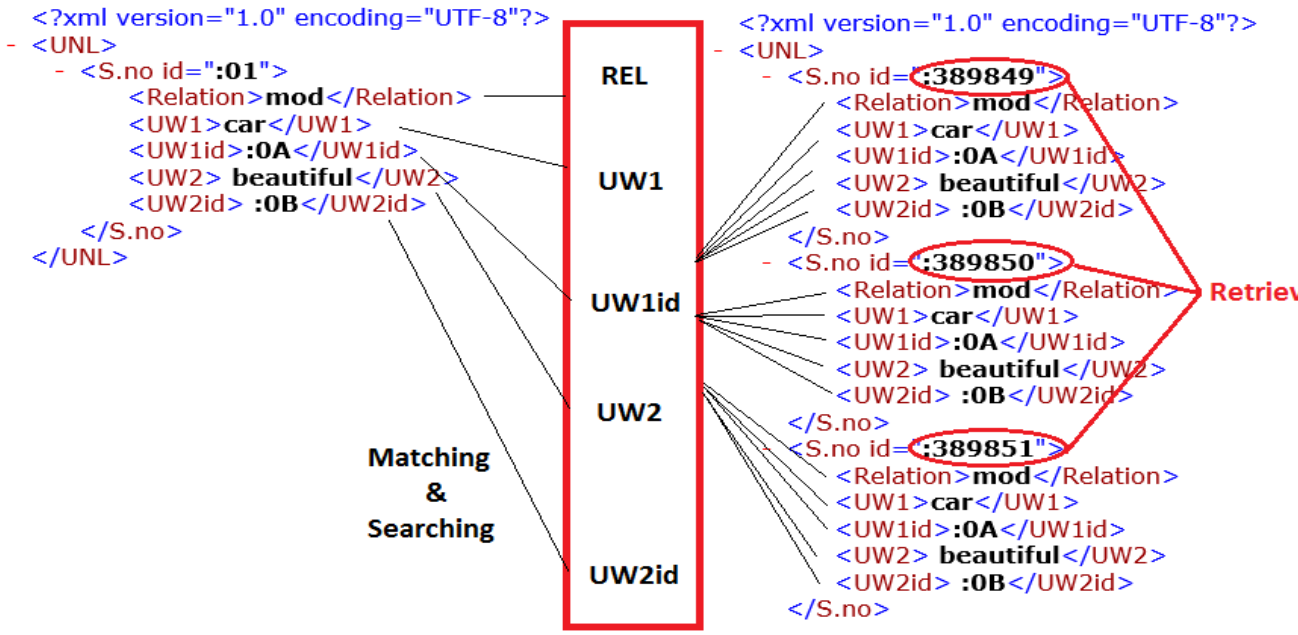


Figure 6.7 Matching and searching of query xml file with sentence in document xml file.

It will result in output as shown in figure 6.8.

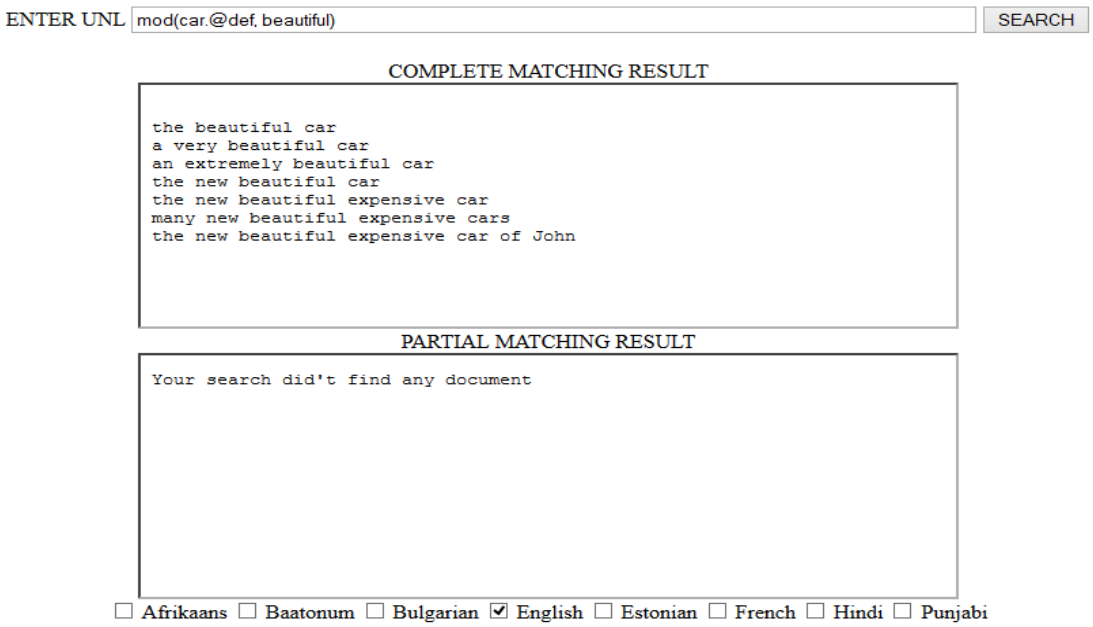


Figure 6.8 Search Module showing Complete matching Search

6.7.2 Partial representation match or Similar RL match

Not only the syntax structure of language involved is different but the translation of two or more different sentences from a language to another may result into same sentence. For example, consider the English sentences below

1) Monkey sat on the wall
2) Monkey sat above the wall
3) Monkey sat over the wall

In above sentences, the spatial preposition "On", "above", and "over" shows the extension of each other. If we translate it in Hindi natural language we could say, **बंदर दीवार पर बैठ गया**, for all the three sentences. So at the same time, monkey could be not only "on" but also "above" and "over" the wall. This brought complexity at additional level.

In the same way, there may be a possibility that two semantically different UNL expressions may analysed from a single sentence in any natural language. For e.g. plc(train.@indef, Patiala.@from) and src(train.@indef, Patiala) are UNL expression

for a single sentence “a train from Patiala” in natural language, English. Although the both UNL analysis inferring a slightly different semantic role but are equally accurate for that natural language sentence.

Since a same sentence in any natural language may be said in many ways by the user. Therefore there may be a possibility that user enter what he understand is correct but the analysis of encoder shows a different meaning from what user need. Again the point is to be noticed is that since our system work on matching relations first. So, by forming a list of relations, as in table 6.3, which may shows similar meaning to universal words, could solve our problem.

Table 6.3 Grouping of similar behaviour UNL relations.

agt,aoj	pur,rsn	cag,cao,cob	coo,dur
fmt,frm,plc,src,to	gol,plt,plc,pof	icl,int,iof	man,met
tim ⟷ tmf and tmt			

Therefore if complete query match failed, then similar relation match searching can be performed by matching all parallel relations and replacing that relation, which is being found by the search engine encoder, with the possible alternatives. There is a major problem with it. It may sometimes return a document which is not much relevant to the document *i.e.* none of the sentence of the document have a query as a sub-graph of it.

To overcome this problem, after getting results we have to parse the results and check the connectivity of query and sentences in that document. In order to establish Connectivity of two edges r1(u1;u2) and r2(u2;u3), we only need to make sure that UW-ID of u2 in first relation is same as that of u2 in the second relation.

As discussed, in this search technique the similar or alternative analyzed relation is found and searched accordingly. Thus this gives another chance to find the precise results. But, recall is still low as in case of complete query search. To overcome this problem, we can introduce a partial query match, which has lower precision but higher recall.

6.7.3 Partial Matching

Discussed in section 3.4.1.3, partial matching has higher recall value of results over complete match results by compromising the precision. Since in partial searching, for a single query more than one UNL sentence could be involved, a separator is there (# in our case) used to separate two UNL sentences.

Let us take an example, "the beautiful book on the table"

Query: "the beautiful book on the table".

UNL: "plc(book.@def, table.@def.@top.@contact)#mod(book.@def, beautiful)".

Explanation: Here, 'book' and 'table' are the UWs, having a relation 'plc' in first UNL sentence.

In the other UNL sentence, that 'book' have 'mod' relation with beautiful. When we perform Complete search, the sentences in the corpus, having all UWs (book, table and beautiful in this case) and respective relation (plc and mod in this case), matched with query UWs and relations respectively, should come up when searching. While when we perform partial search, any one UNL sentence satisfied with query match should come up i.e. "the book on table", "the book on the table about Paris". Although these search results does not have such precision as in complete search but is better to increase recall.

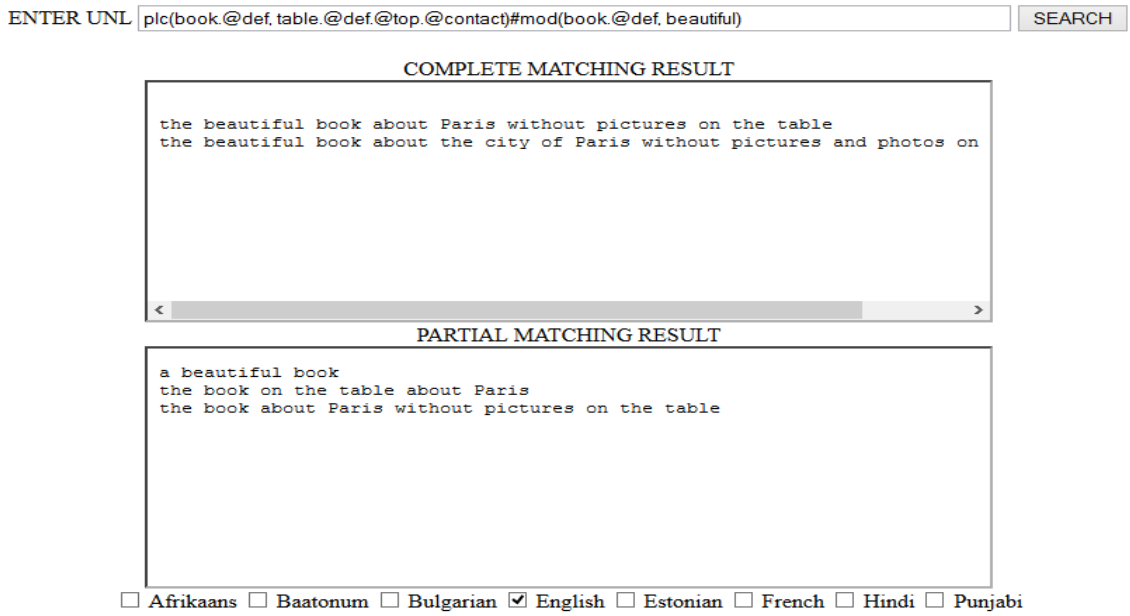


Figure 6.9 Search Module showing Complete matching Search and partial matching result

The core of this module is written in JSP. It just takes as input the query in UNL. Pre-process it as with the same pre-processor used in pre-processing the corpus. Then it calls the Java search program, which returns the results in a raw format. Only the relevant line numbers are returned, as the core is not aware of the language of interface from which it is being called. This parses the results and with the help of corpus module, the relevant lines into the user's language and displays all the results are then available at interface of user according to the user's desired language.

6.7.4 UW Match

The Universal words, which act as headwords, in any UNL expression, are not only the root words but also disambiguate the sense of the word. This technique gives the higher recall value than the keyword matching. To find Rq(d), the relevance of the document d to the query q, for this case, it requires the total number of occurrences of query UWs in the UNL document and the highest number of occurrences of query UWs in any UNL document.

It can be expressed as,

$$\mathrm{Rq(d)} = \frac{\sum_{s \epsilon \mathrm{Sd}} \mathrm{r(s)}}{\mathrm{MaxScore}} \qquad (1)$$

Where,

Rq(d) =relevance of documented to the query q.
r(s) = Total number of occurrences of query UWs in the UNL expression of sentence.
MaxScore = Highest number of occurrences of query UWs in any UNL document.

6.8 Post Processor

In the output produced by the search module, user has links to view the document in different languages. This module is called when user wants to see a document in a particular language. The inputs to this module are the doc id and the language in which to show the document. This module, with the help of corpus translates the UNL document into the language requested by the user and shown to the user interface.

6.9 Interface Module

This module is responsible for interaction with the user and giving him back the results in a user friendly way. This module is implemented entirely in JSP.

There is only one interface for all these languages. When the user enters the query in X language, this interface sends this query to the search module that returns all the relevant information needed by the interface to display the results. The interface will then take this output and then print the output in result area according to the languages chosen by the user.

Figure 6.10 Snapshot of initial interface screen

Chapter 7

Results and discussion

To show the performance of the system, sample queries have been taken to demonstrate the searching results of the system. Since for now, the system works as research work only, the user manually can select the searching technique according to the precision need of the query. The examples are of the 4 cases below

1. Complete query match
2. Partial representation match
3. Partial query match
4. Universal word Match

The corpus, The Encyclopaedia of Life Support Systems (EOLSS), having 12,916 sentences is divided among 13 documents. Each document contains one or more than one chapter. The sentence is relevant to the query if the query search graph of UNL of query matches to the UNL of the sentence in that document.

7.1 Complete Query Match

Sample query: To comprehensive environmental control

UNL of query: to(:01, control(icl>act):3Y)

aoj(comprehensive(aoj>thing):36, control(icl>act):3Y)

UNL graph:

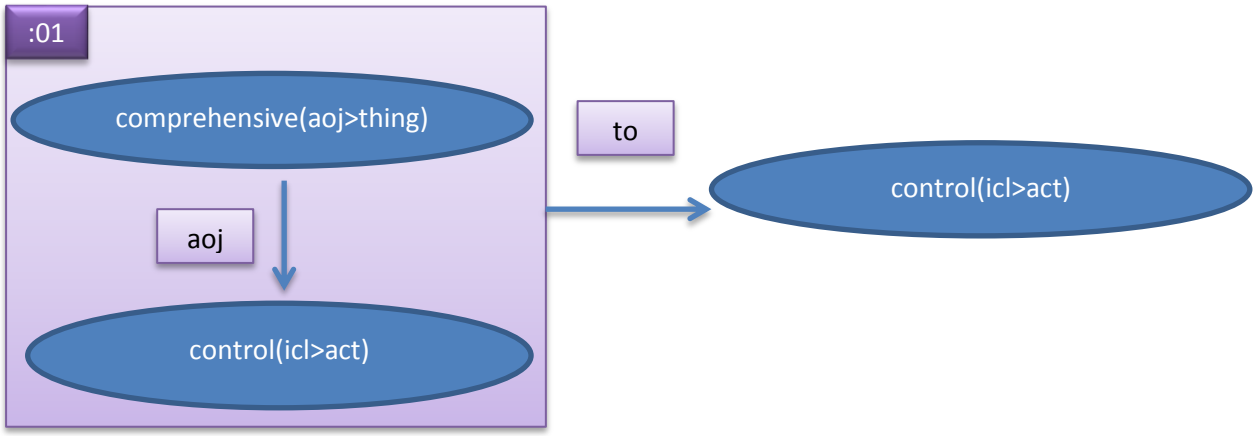


Figure 7.1 UNL graph for the sentence “To comprehensive environmental control”.

Documents Matched and Relevant sentences within it

Match found: 1

Docid: EOLSS2

Relevance: 1

[S:191]

{org:en}

This textbook emphasizes the practical application of sanitary science and engineering theory and principles to comprehensive environmental control.

{/org}

agt(emphasize(icl>give special importance to(agt>thing,obj>thing)):0G.@entry, textbook(icl>book):07.@topic)

…

to(:02, control(icl>act):3Y)

aoj(comprehensive(aoj>thing):36, control(icl>act):3Y)

{/unl}

[/S]

Since the UNL of the sentences of the above document matches the query UNL, proves query graph is sub-graph of document.

7.2 Partial representation match

Sample query: He worked from 1987 to 1999

UNL of query: tim(work(icl>have a job(agt>thing)):0O.@entry.@past, :01)

fmt:01("1999":0F.@entry,"1987":07)

UNL graph:

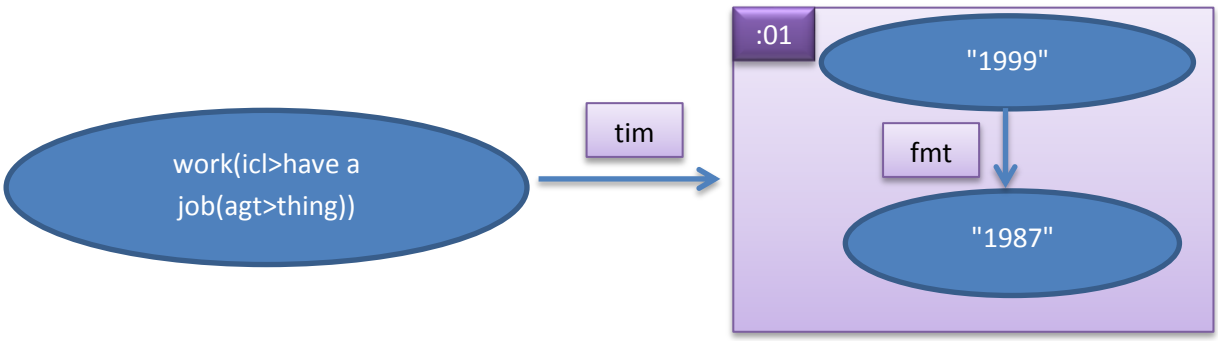


Figure 7.2 UNL graph for the sentence “He worked from 1987 to 1999”.

Documents Matched and Relevant sentences within it

Match found: 1

Docid: EOLSS2

Relevance: 1

[S:295]

{org:en}

From 1987 to 1999, he worked for Japan Water Works Association as Senior Researcher for several fields, which include development studies on advanced water treatment, high-pressure water service, risk management, etc.

{/org}

{unl}

tmf(work(icl>have a job(agt>thing)):0O.@entry.@past,"1999":0F)

tmt(work(icl>have a job(agt>thing)):0O.@entry.@past ,"1987":0M)

…

mod(researcher(icl>person):23, senior(mod<thing):1W)

{/unl}

[/S]

From above search result we can easily predict, how the same natural sentence could be written in many ways. There may be a possibility that the query parser and UNL encoder encode same meaningful sentence in two different ways. This problem has high probability in considering more than one natural language. Since UNL itself provide facility and useful if multiple language involved so we shouldn't compromise with this feature. Partial representation match may be a way to solve this problem. Similar relations i.e. relations targeting same natural sentences, is replaced by each other to getting little high recall. When the UNL sentences of any document matched with replaced UNL query, the result generated.

7.3 Partial query match

Sample query: Sanitation and water supply

UNL of query: and(water supply:2W.@entry, sanitation(icl>equipment):2S)

UNL graph:

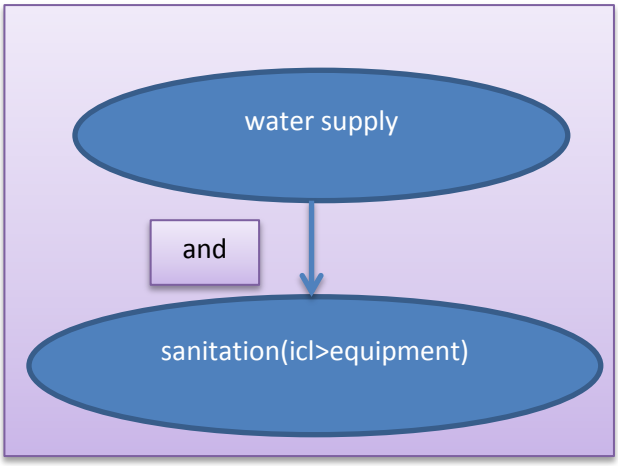


Figure 7.3 UNL graph for the sentence “Sanitation and water supply”.

Documents Matched and Relevant sentences within it

Match found with complete matching: 14

Match found with word “sanitation”: 13

Match found with word “water supply”: 14

Total match found with partial query matching: 27(Shown here 4)

[S:90]

{org:en}

The Global Water Supply and Sanitation Assessment 2000, a joint monitoring program by WHO and UNICEF.

{/org}

{unl}

aoj(program(icl>plan):2C.@entry.@indef, :01)

mod:01(supply(icl>act):0M.@def, water(icl>liquid):0G)

…

mod:01(assessment(icl>act):1C.@entry, sanitation(icl>equipment):11)
and:01(assessment(icl>act):1C.@entry, supply(icl>act):0M.@def)
…
aoj:01(global(icl>overall(aoj>thing)):09, supply(icl>act):0M.@def)
{/unl}
[/S]

[S:19]
{org:en}
Constrains to improving water and sanitation services
{/org}
{unl}
obj(constrain(agt>thing,obj>thing):02.@entry, improve(agt>thing,obj>thing):0G)
and:01(sanitation(icl>equipment):0Y.@entry, water(icl>liquid):0O)
mod(service(icl>system):19.@pl, :01)
obj(improve(agt>thing,obj>thing):0G, service(icl>system):19.@pl)
{/unl}
[/S]

[S:148]
{org:en}
Appraisal of Water Supply Rehabilitation Project in the Republic of the Philippines.
{/org}
{unl}
obj(appraisal(icl>judgement):02.@entry, project(icl>plan):17)
…
mod(project(icl>plan):17, water(icl>liquid):0F)
plc(project(icl>plan):17, republic(icl>country):1M.@def)
mod(rehabilitation(icl>act):0S, supply(icl>act):0L)
{/unl}
[/S]

[S:294]

{org:en}

He was responsible for appraisal and evaluation of bank-financed loan projects in the water supply, sewerage and sanitation sectors.

{/org}

{unl}

aoj(responsible for(aoj>thing,obj>thing):09.@entry.@past, he:02.@topic)

…

and:02(sewerage(icl>system):2U, water supply:2G)

…

plc(project(icl>plan):20.@pl, sector(icl>activity):3I.@def.@pl)

and:02(sanitation(icl>equipment):37.@entry, sewerage(icl>system):2U)

{/unl}

[/S]

Partial search results may be combined with complete search results to increase recall of the search results. The results comes only with partial query match have less precision than the results comes up with complete search. So if the results are to be sorted according to the relevancy, the purely partial search results come after the search results of complete query match. To make it clear the search results, both the search results i.e. complete query search and partial query search, has been shown in different windows of the web page. Partial query search is equally important to that of the complete query search since the user may be satisfied with the document having some of the information that required.

7.4 Universal word Match

Sample query: marketing of fruits and vegetables

UNL of query:

mod (*marketing(icl>commerce)*:00.@entry, :01)

and:01(*vegetable(icl>food>functional thing,icl>plant>livingthing)*:0O.@entry.@pl,

fruit(pof>plant):0D.@pl)

Matched UNL documents and relevant sentences therein: Total match found: 8 (shown two here)

docid: EOLSS5 , relevance: 1

[S:5]

…

aoj(promote(icl>support(agt>thing,obj>thing)):0M.@entry.@present.@progress,government(icl>governmental organization):04.@def)

plc(promote(icl>support(agt>thing,obj>thing)):0M.@entry.@present.@progress,country(icl>region):2K.@def)

obj(promote(icl>support(agt>thing,obj>thing)):0M.@entry.@present.@progress, *marketing(icl>commerce):16*)

mod(*marketing(icl>commerce):*16, organized(aoj>thing):0W)

mod(*marketing(icl>commerce):*16, commodity(icl>goods):1W.@pl)

mod(commodity(icl>goods):1W.@pl, agricultural(mod<thing):1J)

…

[/S]

[S:8]

…

mod:05(commodity(icl>goods):2N.@pl, agricultural(mod<thing):2A)

and:04(*marketing(icl>commerce):*1E.@entry, :03)

and:03(consumption(icl>phenomenon):0Y.@entry, production(icl>product):0J.@def)

…

[/S]

[S:13]

…

mod(*marketing(icl>commerce):*4W, produce(icl>food):6P)

mod(produce(icl>food):6P, :01) mod(produce(icl>food):6P, agricultural(mod<thing):6C)

and:01(vegetable(icl>food):5S.@entry.@pl, *fruit(pof>plant):5H*.@pl)

…

[/S]

docid: EOLSS6 , relevance: 0.888889

[S:601]

…

and:01(vegetable(icl>food):11.@entry.@pl, *fruit(pof>plant):*0Q.@pl)

…

[/S]

[S:603]

…

mod(*marketing(icl>commerce):*04.@def, :01)

and:01(vegetable(icl>food):0W.@entry.@pl, *fruit(pof>plant):*0L.@pl)

…

[/S]

[S:608]

…

mod(*marketing(icl>commerce):*04.@def, :01)

and:01(vegetable(icl>food):0W.@entry.@pl, *fruit(pof>plant):*0L.@pl)

…

[/S]

In the above, the bold, italicized text represents the Universal Words in the sentence, which match with the query.

7.5 Comparative Analysis

Comparative analysis is done to check the performance according to the relevancy and precision of each search strategy. Out of 20 sentences searched, all show same kind of sequence *i.e.* as we go down according to the fall model discussed above, the precision shows decreasing behaviour and recall increases. One example is discussed.

Sample Query: **primary pure water processing system**

UNL of the Query:

mod(system(icl>equipment):0W.@entry,primary(mod>thing):02)

obj(processing(icl>act):0L,:01)

aoj:01(pure(icl>not mixed(aoj>thing)):0A,water(icl>liquid):0F)

mod(system(icl>equipment):0W.@entry,processing(icl>act):0L)

UNL graph:

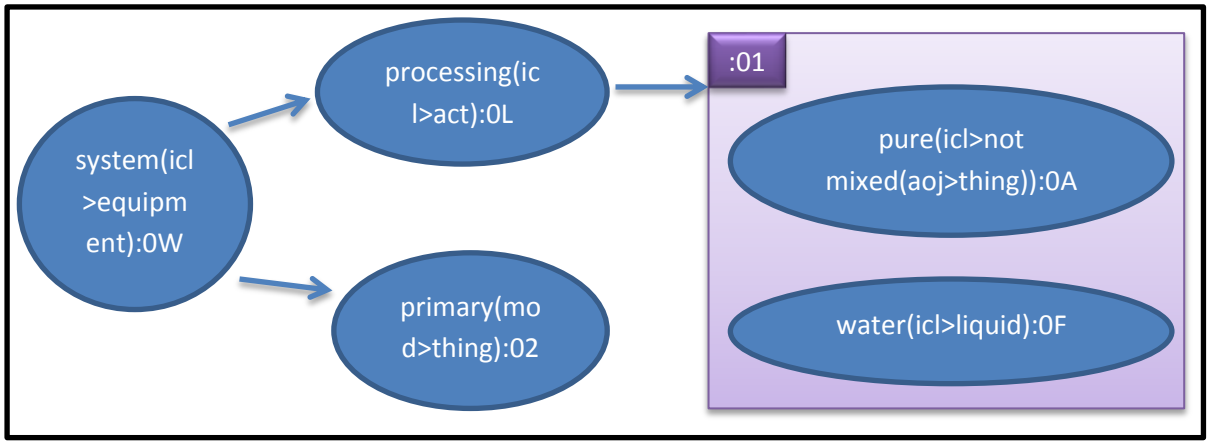


Figure 7.4 UNL graph for the sentence "primary pure water processing system".

TABLE 7.1 RESULTS TO SHOW COMPARATIVE RELEVANCY OF SENTENCES COMES WITH DIFFERENT STRATEGIES

Strategy performed	**No. of match found**	**Example sentence**	**DocumentId (no.of matches found)**	**Relevance**
Complete query match	3	the primary pure water processing system for removing dissolved gases.	Eolss76(3)	1,1,1
Partial representation match	3	Pure water processing system can be primary or secondary	Eolss76(3)	1,0.87,0.87
Partial query match	16	The source of water has a major effect on water processing system	Eolss75(4), Eolss76(10), Eolss84(1), Eolss85(1),	0.786, 0.953, 0.196, 0.196, 0.196, 0.392

		design and hence costs.	Eolss87(1), Eolss113(2)	
UW Match	27	primary pure liquid processing system	many documents matched	low
Keyword match	8560	water is pure	many documents matched	lowest

Above table proves that, when we performed Searching in our search engine, it provides results with higher precision with complete query match and highest recall with keyword match search strategy. The results in nut shell are shown in table 1, with some of the example to show what kind of result sentences, with document id, are comes up when above query is passed.

Chapter Summery

All searching strategies have been tested and briefly summarized in this chapter. Complete search shows results with precision 1 *i.e.* the results found are fully matched with the query, while other search strategies shows precision decreasing as we go down with the fall model. To increase recall universal searching is introduced. Like keyword match it shows many results but have higher precision.

Chapter 8

Conclusion and Future scope

8.1 Conclusion

Most of the traditional search engine on web, searches on meaningless tokens provided by user. Although many meaning based search engines are evolved but restricted to either a single language or have very few applications. Cross-lingual communication is needed to shorten the barrier between languages. This all features can only possible by using Universal Networking Language (UNL).

UNL is an intermediate language which is language independent also. It captures the relationship between universal words and their attribute. Hence cross-lingual and meaning based properties could be achieved together rather than using different language translators.

Since previous enconverter (ENCO) and deconverter (DECO) are not very successful for all languages to generate UNL sentences. Thus, IAN and EUGENE framework can be used as enconverter and deconverter.

As the content of the web is exponentially expanded, hence created a bottleneck, there is a need of precise and relevant search results. Hence complete search is being proposed in this report. Also to increase some amount of recall, partial searching is being proposed. Thus using above searching techniques, enconverter IAN and with the help of UNL, a knowledge based semi-automated cross-lingual meaning based search engine has been developed.

8.2 Future scope

In future following areas will touch upon:

- It is a semi-automated system, in future IAN API can be included to get automated system.
- Strong Crawlers and HTML parsers can be included.
- Scalability of the search techniques can be tested on bigger corpus.

Appendix-A

Some UNL interpretation of EOLSS.

[S:2]
{org:en}
Health problems and their resolution
{/org}
{unl}
and(resolution(icl>solution):0S.@entry, problem(icl>abstract thing):09.@pl)
mod(problem(icl>abstract thing):09.@pl, health(icl>state):02)
mod(resolution(icl>solution):0S.@entry, they(icl>thing):0M.@pl)
{/unl}
[/S]

[S:3]
{org:en}
Yasumoto Magara
{/org}
{unl}
mod(Magara(iof>Japanese):0B.@entry, Yasumoto(iof>Japanese):02)
{/unl}
[/S]

[S:321]
{org:en}
Due to the diversity of the natural forms of the existence of water and diversity of forms of water using by humans (biological and technical), the vast multiplicity of water properties was explored (physical, chemical, biological, and technological).
{/org}
{unl}
obj(due to(aoj>thing,obj>thing):02.@entry, :01:01)
agt:01(use(agt>thing,obj>thing):2U.@progress, human(icl>living thing):33.@pl)
mod:01(form(icl>state):2F.@pl, water(icl>liquid):2O)

aoj:01(natural(aoj>thing):0U, form(icl>state):12.@def.@pl)
agt:01(use(agt>thing,obj>thing):2U.@progress, human(icl>living thing):33.@pl)
mod:01(form(icl>state):2F.@pl, water(icl>liquid):2O)
aoj:01(natural(aoj>thing):0U, form(icl>state):12.@def.@pl)
mod:01(diversity(icl>quality):22.@entry, form(icl>state):2F.@pl)
mod:01(form(icl>state):12.@def.@pl, existence(icl>state):1F.@def)
and:01(diversity(icl>quality):22.@entry, diversity(icl>quality):0D.@def)
mod:01(diversity(icl>quality):0D.@def, form(icl>state):12.@def.@pl)
mod:01(diversity(icl>quality):0D.@def, form(icl>state):12.@def.@pl)
obj:01(use(agt>thing,obj>thing):2U.@progress, water(icl>liquid):2O)
obj:01(existence(icl>state):1F.@def, water(icl>liquid):1S)
{/unl}
[/S]

[S:805]
{org:en}
The boundaries and scales of the two categories require updating continuously in the light of the development of scientific and technical knowledge and the application of new technologies.
{/org}
{unl}
agt(require(icl>demand(agt>thing,obj>thing)):1E.@entry, :01)
and:01(scale(icl>attribute):0L.@entry.@pl.@topic,
boundary(icl>line):06.@def.@pl.@topic)
and:03(application(icl>act):45.@entry.@def, development(icl>activity):2S.@def)
man(update(agt>thing,obj>thing):1M.@progress, continuously:1V)
agt(require(icl>demand(agt>thing,obj>thing)):1E.@entry, :01)
and:01(scale(icl>attribute):0L.@entry.@pl.@topic,
boundary(icl>line):06.@def.@pl.@topic)
and:03(application(icl>act):45.@entry.@def, development(icl>activity):2S.@def)
man(update(agt>thing,obj>thing):1M.@progress, continuously:1V)
obj:03(development(icl>activity):2S.@def, :02:02)
man(require(icl>demand(agt>thing,obj>thing)):1E.@entry,
update(agt>thing,obj>thing):1M.@progress)

qua:01(category(icl>group):13.@def.@pl, "2":0Z)
aoj:03(new(icl>recent(aoj>thing)):4K, technology(icl>knowledge):4O.@pl)
man(require(icl>demand(agt>thing,obj>thing)):1E.@entry, in the light of(obj>thing):28)
mod:01(scale(icl>attribute):0L.@entry.@pl.@topic, category(icl>group):13.@def.@pl)
man(require(icl>demand(agt>thing,obj>thing)):1E.@entry, in the light of(obj>thing):28)
mod:01(scale(icl>attribute):0L.@entry.@pl.@topic, category(icl>group):13.@def.@pl)
obj:03(application(icl>act):45.@entry.@def, technology(icl>knowledge):4O.@pl)
obj(in the light of(obj>thing):28, :03)
and:02(knowledge(icl>information):3N.@entry, scientific(aoj>thing):37)
{/unl}
[/S]

Appendix-B

Some UNL interpretation of UGO-A1.

Languages involved:

Afrikaans(af), Baatonum(bba), Bulgarian(bg), English(en), Estonian(et), French(fr), Hindi(hi), Punjabi(pa).

[S:389653]
{org:en}
that book
{/org}
{af}
daardie boek
{/af}
{bba}
tire tè
{/bba}
{bg}
онази книга
{/bg}
{bg}
книгата до теб
{/bg}
{bg}
книгата до вас
{/bg}
{et}
too raamat
{/et}
{fr}
ce livre
{/fr}
{hi}

उस किताब

{/hi}

{hi}

वह किताब

{/hi}

{pa}

ਉਹ ਕਿਤਾਬ

{/pa}

{unl}

[W]

book.@medial

[/W]

{/unl}

[/S]

REFERENCES

[1] The Universal Networking Language Specifications, Version 3, Edition 1, UNL Center, UNDL Foundation, www.unl.ias.unu.edu/unlsys/unl/UNL%20Specifications.htm

[2] "Universal Networking Language (UNL): History of UNL," [Online]. Available: http://www.unlweb.net/unlweb/index.php?option=com_content&view=article&id= 58:unl&catid=54:unlweb [Accessed 17 April 2014].

[3] S. Alansary et al., "A Suite of Tools for Arabic Natural Language Processing: A UNL Approach," in IEEE Transactions, vol. 33, 2013 pp. 56-99.

[4] A.Martins, G. Tissiani and R. M. Barcia, "A Framework for the Development of Universal Networking Language E-Learning User Interface," in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 268-275.

[5] P. Kumar, "UNL Based Machine Translation System for Punjabi Language," Ph.D. dissertation, Dept. Comput. Sci. Eng., Thapar University, Patiala, 2013.

[6] P. Kumar and R. K. Sharma, "Generation of UNL Attributes and resolving relations for Punjabi Enconverter," *Malaysian Journal of Computer Science*, vol. 24, no. 1, pp. 34-36, 2011.

[7] "History of UNL," [Online]. Available: http://www.unlweb.net/wiki/Introduction_to_UNL#History. [Accessed 20 Dec 2013].

[8] "Universal Networking Language (UNL)," [Online]. Available: http://www.undl.org/index.php?option=com_content&view=article&id=46&Ite

mid=63&lang=en. [Accessed 11 Dec 2013].

[9] M. Monju, T. Shilpa, D. Smita and P. Bhattacharya, ”Knowledge extraction from Punjabi Text,” in Knowledge Based Computer Systems, M. Sasikumar, D. Rao and P. R. Prakash, Mumbai: Allied Publishers, 2000, pp.193-204.

[10] " Universal Networking Language: UNL development," [Online]. Available: http://dev.undlfoundation.org/index.jsp.[Accessed 12 May 2014].

[11] "Universal Networking Language: Universal Words," [Online]. Available: http://www.undl.org/publications/UW%20and%20UNLKB.htm. [Accessed 15 May 2014].

[12] S. Ermolaev, “My Invention—Intelligent Semantic, Clever, meaning based Searching System”.

[13] S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine”, Computer Networks and ISDN Systems 30 (1 –7), 1998.

[14] http://www.oingo.com,http://www.excite.com, http://www.simpli.com

[15] D. Tümer, M. A. Shah, and Y. Bitirim, “An Empirical Evaluation on Semantic Search

Performance of Keyword-Based and Semantic Search Engines: Google, Yahoo, Msn and

Hakia”, 2009 4th International Conference on Internet Monitoring and Protection (ICIMP ’09) 2009.

[16] G. Sudeepthi, G. Anuradha, and M. Surendra Prasad Babu. "A Survey on Semantic Web Search Engine." *International Journal of Computer Science* 9 (2012).

[17] X. Zhang, “MIRTH --Chinese/English Search Engine: A Multilingual Information Retrieval Tool Hierarchy For World Wide Web ‘Virtual Corpus’ and Training Resource in Computing and Linguistics & Literature,” UK, 1996.

[18] G.Erbach, G.Neumann, and H.Uszkoreit, “MULINEX: Multilingual Indexing, Navigation and Editing Extensions for the World Wide Web”. 1997, published in Hull, D. and Oard, D. eds. Cross-Language Text and Speech Retrieval - Papers from the 1997 AAAI Spring Symposium. AAAI Press, Stanford.

[19] A. F. Okumura and H. Takagi, “Regional language search engine developed,” http://www.ciol.com/ciol/news/33094/regional-language-search-engine-developed, November 6, 2004.

[20] P. Buitelaar, K. Netter and F. Xu, “Integrating Different Strategies for Cross-Language Information Retrieval in the MIETTA Project”, In Proceedings of TWLT14, Enschede, the Netherlands, December 1998. http://www.mietta.info/docs/mietta-twlt.pdf

[21] M. Surve, S. Kagathara, P. Bhattacharyya and Agro Explorer Group, “Agro Explorer: a Meaning Based Multilingual Search Engine”, In Proceedings of the International Conference on Digital Libraries (ICDL) 2004, Volume 2, New Delhi, India. http://www.mlasia.iitb.ac.in/docs/agro_icdl.pdf Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 167-174.

[22] J. M. Pellizoni, “Flexibility, Configureurability and Optimality in UNL Deconversion via Multiparadigm Programming,” in Universal Networking Language: Advances in Theory and Applications, vol. 12. C. Jesus, G. Alexander and T. Edmundo, Mexico: Centre for Computing Research of IPN, 2005, pp. 175-194.

[23] " UNDL Organization:UNL Explorer," [Online]. Available: http://www.undl.org/unlexp.[Accessed 23 March 2014].

[24] " Google," [Online]. Available: http://www.google.com.[Accessed 04 April 2014].

[25] " Universal Networking Language: UNLarium," [Online]. Available: http://www.unlweb.net/unlarium/index.php.[Accessed 05 April 2014].

[26] K. H. Johansson, J. Lygeros, and S. Sastry. "Encyclopedia of life support

systems (EOLSS)." *H. Unbehauen, Ed* (2004).

[27] “Universal Networking Language (UNL): Tools,” [Online]. Available: http://www.unlweb.net/wiki/Tools [Accessed 28 March 2014].

[28] “Universal Networking Language (UNL): Login,” [Online]. Available: http://www.unlweb.net/user/index.php?page=login[Accessed 29 March 2014].

- CUP250 - Certificate of Proficiency in UNL is a certificate issued by the UNDL Foundation.
- CUP500- Certificate of Proficiency in UNL is a certificate issued by the UNDL Foundation.
- CLEA250- Certificate of Language Engineering Aptitude in UNL is a certificate issued by the UNDL Foundation.
- CLEA500- Certificate of Language Engineering Aptitude is a certificate issued by the UNDL Foundation.
- CLEA750- Certificate of Language Engineering Aptitude is a certificate issued by the UNDL Foundation.
- CLEA1000- Certificate of Language Engineering Aptitude issued by the UNDL Foundation.

Research Publications

Research Paper Accepted

- Shivangi Nanda, Parteek Bhatia, "Semantic Search Engine and its Strategies with IAN Encoder" in International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) (IEEE), ISBN No. 978-1-4799-3914-5/14/$31.00,2014,IEEE.