UNL Punjabi Deconverter

A thesis
submitted in partial fulfillment of the requirements
for the award of degree of

Master of Engineering
in
Software Engineering

Under the Supervision of
Mr. Parteek Bhatia
Lecturer
CSED, TIET, Patiala.

Submitted By
Anjuman Chawla
(Roll No 8043102)

Computer Science & Engineering Department
Thapar Institute of Engineering & Technology
(Deemed University), Patiala-147004

May 2006

Candidate's Declaration

I hereby certify that the work which is being presented in the thesis entitled, **"UNL Punjabi Deconverter",** submitted by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering at Computer Science & Engineering Department of Thapar Institute of Engineering & Technology (Deemed University), Patiala, is an authentic record of my own work carried out under the supervision and guidance of Mr. Parteek Bhatia.

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

Anjuman Chawla

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

Mr. Parteek Bhatia
Lecturer
Computer Science & Engineering Department
Thapar Institute of Engineering & Technology
Patiala- 147004

Countersigned by:

Dr.(Mrs) Seema Bawa
Head
Computer Sc. & Engg. Department
Thapar Institute of Engg. & Technology
Patiala- 147004

Dr. T. P. Singh
Dean of Academic Affairs
Thapar Institute of Engg. & Technology
Patiala- 147004

Acknowledgement

I wish to express my deep gratitude to Mr. Parteek Bhatia, Lecturer, Computer Science & Engineering Department for providing his uncanny guidance and support throughout the thesis work and preparation of the thesis report.

I am also thankful to Dr. (Mrs.) Seema Bawa, Head, Computer Science & Engineering Department, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank all the staff members and my co-students who were always there at the need of the hour and provided with all the help and facilities, which I required for the completion of the thesis.

At last but not the least I would like to thank God and mine parents for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Anjuman Chawla

(8043102)

Abstract

The World Wide Web represents a formidable tool for communication and information access. With simple equipment, it is possible to access innumerable documents about a huge variety of topics, from any place around the world. However, despite the abundance of information, languages very often cause problems. When most of the web pages today are written in few most commonly used languages like English, French, Chinese etc, it becomes difficult for a person with insufficient knowledge of these languages to access and use this tool of communication and information. This has prompted the need to devise means of automatically converting the information from one natural language to another natural language, called Machine Translation. This process needs syntactic and semantic analysis of both source and target languages. Interlingua based machine translation has received a considerable attention because of economy of translation of effort and also additional attraction of the Interlingua providing a knowledge representation scheme.

In this thesis work, we have dealt with the language independent deconverter for the Punjabi language it takes as input a UNL (Universal Networking language) expression. For the purpose of conversion we use Interlingua which follow the UNL specifications proposed by UNU/IAS Tokyo. UNL (Universal Networking language) is a language used to represent a semantic graph equivalent of a concept (contained in text document). The system takes a set of UNL expression as input and with the help of language independent algorithm and language dependent data generates corresponding Punjabi sentence. The process of deconversion involves syntax planning, case marker generation and morphology phase. The syntax planning phase is aimed at generation of proper sequence of words for the target sentence. These phases first reads the input UNL file and convert it into semantic-net like structure known as nodenet. Nodenet is a directed acyclic graph structure, which defines the sentence in the form of Directed Acylic Graph. We use lexicon files to map the UWs to target language worlds. After generating a nodenet, the

problem of the syntax plan generation get reduce to the problem of Directed Acylic Graph traversal. Proper traversal of the node net generates the syntax plan of the target sentence. This syntax plan needs to be processed by the case-marking file, which apply proper case marker for each and every relations. This case-marking phase is next processed by the morphology phase. The morphology phase gives a final form of the target sentence.

Contents

Table of Contents

List of Figures

Chapter 1: Introduction

The mechanization of translation has been one of humanity's oldest dreams and this thesis just an attempt to fulfill it. Machine translation (MT), also known as "automatic translation" or "mechanical translation," is the name for computerized methods that automate all or part of the process of translating from one human language to another. MT is a multi-disciplinary field of research as it incorporates ideas from linguistics, computer science, artificial intelligence, statistics, mathematics, philosophy and many other fields.

Language is a medium of understanding others and making oneself understood by others. All over the world, people speak various languages. In India itself there are 18 recognized language and 6000 dialects. But Internet has bought a revolutionary change. Internet has made world a smaller place. However language is a major barrier. Documents written in one language remain locked up for the people who don't know the language. Out of the nine hundred million population of India, nearly one thousand is currently excluded from participating actively in this so called information age.

Punjabi is one of the major languages spoken in India. So the problem of Punjabi language generation has its own importance. It is impossible for human to manually translate huge number of documents. This give rise to machine translation, a wide research area in AI. . Through this thesis, we have attempted to design a deconverter i.e. generator system for the Punjabi language

1.1 Language Translation

In order to overcome the language barrier, many attempts have been made in the past. Professional translators have been bridging such a communication gap. The quantity of translation by human however is rather small as compared to the required communication needed for the different languages. The main reason for such a limitation is high cost involved in the translation work. In addition, the number of translators for minor

language is rather small. Translation made by human being, thus has its limitation in terms of cost and human resources

Computer translation systems have made significant progress. Some of them are now being incorporated in network browsers. The demand for these systems indicates how large the language problem is among Internet users. Computer translation systems are useful under limited conditions. For instance, the user can evaluate and modify a translated document in his own language, but seldom in the other language. However, after translating with a computer, the user has to work to edit the output document. In addition it would require language knowledge to edit the translation of the document in the other language. In sending information throughout the world, the sender normally does not know the language of the recipient. In this case, the sender is bound to use a computer translation system blindly, because he cant check whether the translated results are correct or not. This is a serious limitation in current computer translation system, and explains in parts, their limited acceptance.

1.2 UNL Project

To give the world a gift a millennium, UNU/IAS (United Nations University/Institute of Advanced Studies) started the UNL (Universal Networking Language) project .UNL project is aimed at elimination of the language barrier. Main approach of this system is to represent information in the form of knowledge, using language independent interlingua to represent knowledge. With this characteristic in mind, Universal Networking Language (UNL) is developed.

UNL intermediates understanding among different natural languages. UNL represent sentences in the form of logical expression without ambiguity. It is an intermediate language, which allows communication among people of different language using their mother tongue. The UNL is a language specification for the exchange of information over the Internet. The motivation behind UNL is to devloped an interlingual representation such that semantically equivalent sentences of all language have the same interlingual representation. UNL plays the role of an interface between different languages to exchange information. UNL represent each sentence in the given text as set of relation.

The UNL vocabulary consists of

- **Universal Word**: represent the word meaning
- **Relation Label**: represent the relation between UWs
- **Attribute Label**: represent the further definition or additional information, which appears in the sentence.

For example,

The English sentence is *Girls eat mangoes.*

The corresponding UNL expression will be

[s]

agt(eat(icl>event}.@entry.@present, girl(icl>person).@pl)

obj(eat(icl>event}.@entry.@present, mango(icl>food).@pl)

[/s]

This UNL expression can be translated back to other target language. Thus, UNL is an intermediate language, which allows communication among people of different languages, using their mother tongue. The UNL system will transform natural languages to UNL expression (using Enconverter) and UNL expression to natural language (using Deconverter).

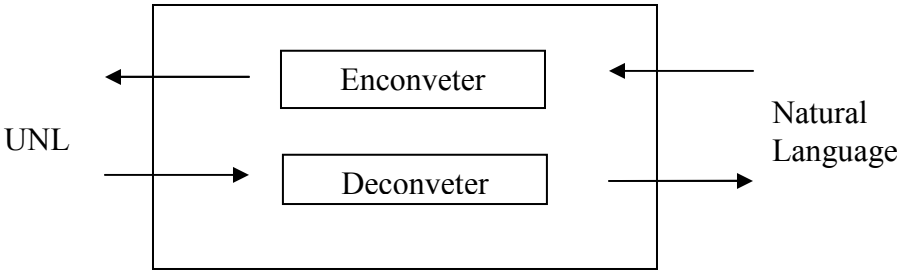


Figure 1.1 UNL System

1.3 Language Independent Generator

As a part of the UNL system, we have designed independent generator for Punjabi language. This engine takes UNL expression as input and generates target language (Punjabi) sentence with the help of various database files like lexicon files and morphological rule files.

The deconverter can be logically portioned into three phases as:

1.) Syntax planning phase
2.) Case marking phase
3.) Morphology phase

The syntax planning phase is aimed at generation of proper sequence of words for the target sentence. This phase first reads the input UNL file and converts it into semantic-net like structure known as nodenet. Nodenet is a directed acyclic graph structure, which defines the sentence in the form of DAG (Dirested Acylic Graph). We use lexicon files to map the UWs to target language worlds. After generating a nodenet, the problem of the syntax plan generation get reduce to the problem of DAG traversal. Proper traversal of the node net generates the syntax plan of the target sentence.

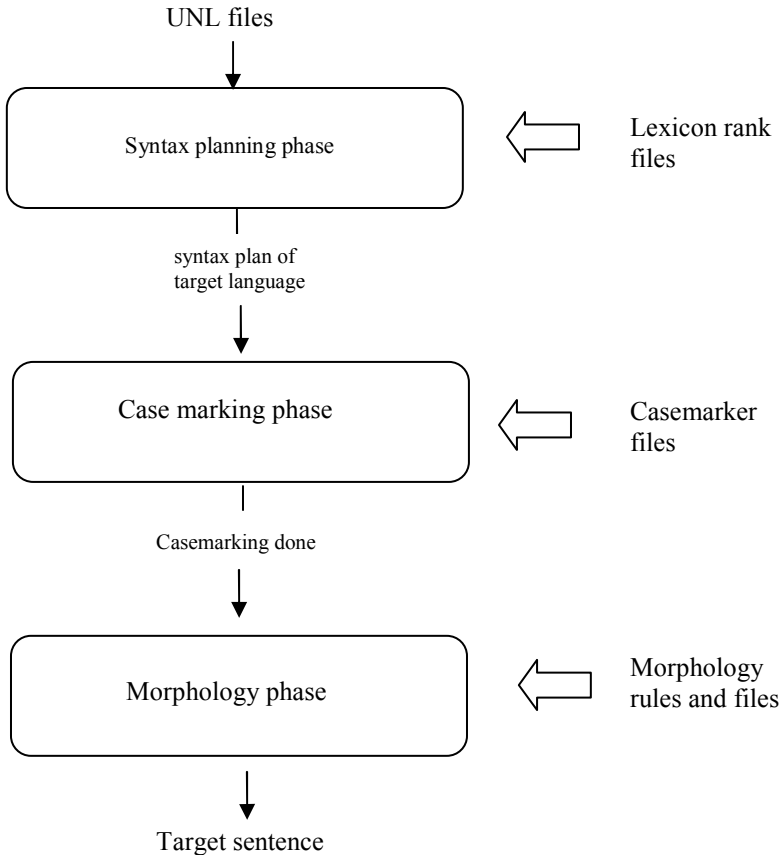


Figure 1.2 Language Deconverter System

The syntax planning phase generates the sequence of words, which cannot express the complete contents of the sentence. This syntax plan need to be processed by the case

marking file, which apply proper case marker for each and every relations. This case marking phase is next processed by the morphology phase. The morphology phase gives a final form of the target sentence.

1.4 Motivation of the project

An effort toward building a language independent Deconverter System has been going on at IIT, Bombay. This system was demonstrated for the UNL to Hindi and UNL to Marathi. With the motivation from this, we started to deign a system for Punjabi languge too. The complete system is redesigned for the Punjabi language requirement and is explicitly dedicated to achieve language independence.

Chapter2: Natural Language Generation and Machine Translation

2.1 Natural Language Generation

Natural language generation is a sub field of artificial intelligence and computational linguistics that is concerned with the construction of computer system that can produce understandable texts in English or other human language from some underlying non-linguistic representation of information [5].

NLG systems combine knowledge about language and the application of domain to automatically produce documents, reports, explanation, help message and other kinds of texts.

2.1.1 Applications of Natural Language Generation

The most common use of natural generation technology is to create computer system that present information to people in a representation that they find easy to comprehend. Internally, Computer systems use representation, which are straightforward for them to manipulate, such as databases, accounting, spreadsheets, expert system's knowledge bases, simulations of physical systems etc. These representations of information require a considerable amount of expertise to interpret. This means there is often a need for systems that can present data in an understandable form to non-expert users.

The important domains for the application of natural language generation are

- To generate textual weather forecast from representations of graphical weather maps
- To automatically generate documents
- To summarize statistical data extracted from a database or spreadsheet
- To explain medical information in a patient friendly manner
- Machine translation between natural languages
- Translation from a source representation to multiple natural languages.

2.2 Machine Translation

Machine translation (MT), also known as "automatic translation" or "mechanical translation," is the name for computerized methods that automate all or part of the process of translating from one human language to another. MT is a multi-disciplinary field of research as it incorporates ideas from linguistics, computer science, artificial intelligence, statistics, mathematics, philosophy and many other fields [11].

2.2.1 Need for Machine translation

Today the need of translation is much more as in past. Here are some important requirements for the Machine translation system.

- The Internet changes the world very fast. Now we can find vast amount of knowledge on Internet. But most of this information is in English. In the context of rural India, most of this information is effectively unavailable to the rural masses that are not qualified for English. In spite of all the progress that is being made in the field of Information Technology, rural masses remain deprived of the technological advancements. The one of the primary reasons for this is the incapability in information distribution and language barrier is one of the biggest hurdles in this information distribution. There is a great demand to translate Web pages and electronic mail messages [4].
- There is also a demand of Internet-based online translation services. In this direction, the pioneer was the service offered in France by Systran.
- Demand of online versions of electronic dictionaries as 'translation systems' to help human translators for translation.
- The Internet also suggests other roles for MT. Searching data and information in languages unknown or poorly known by the user is a formidable obstacle. "Cross Language Information Retrieval' (CLIR) systems, are designed in this direction, which enable queries expressed as keywords in one language to be translated into keywords of another language and for searches to be conducted on databases in multiple foreign languages.

- The results of MT research could impact major aspects of life, including politics, culture, science, philosophy, and business. If MT can become accurate and efficient enough, it can break down cultural barriers and make communication between speakers of different languages much easier.
- Commercially, MT can allow companies to translate product manuals more quickly into the target language or languages. Thus, MT systems can expand a company's market, save translators time and companies money in the process of translation.

So we can easily conclude that there is a huge market for translation from one language to another or multi-lingual translation systems.

2.2.2 Challenges for Machine Translation

Languages are challenging to translate for several reasons, and some of the obstacles are presented here. Major issues in MT involve ambiguity, structural differences between languages, and multiword units like idioms [25].

Ambiguity: Languages can present ambiguity on several levels. If a *word* can have more than one meaning, it is classified as *lexically* ambiguous. An approach to solve this problem is syntactic parsing or statistical analysis. For Example:

(a) I am drinking water.

(b) Do not water the plants.

In this case *water* word is interpreted as a noun in (a) and in (b) it is interpreted as verb.

When a phrase or sentence can be interpreted in more than one way, it is called **structurally ambiguous**. It is especially challenging because it requires a deep understanding of the speaker's intention, and we can often not be certain of what exactly the speaker meant. For Example:

(c) We killed that team last night

This sentence can be interpreted in two ways: first, "we ended the lives of that team" and second, "we beat that team in quite a resounding manner."

Structural and lexical differences between languages: Word orderings often differ between languages. For example English language is Subject-Verb-Object (SVO)

language whereas Punjabi language is Subject-Object-Verb (SOV) language. Moreover articles are used in English language whereas there are no articles in Punjabi Language.

Idioms, Articles and Phrases: A group of words or collocations such as Idioms or Phrases cannot be translated with the normal rules used for MT. An idiom like "To kill two birds with one stone" in English translated literally would make absolutely no sense in Punjabi language.

Articles may be used in a language in one context, but in the target language, articles might not be used. *Lexical holes* exist where the target language has to represent a word in the source language with a phrase because there is no exact translation.

Tense Generation: *Tense generation* is another structural difference problem. Tenses may exist in one language but not another. For example: a language like English has explicit present progressive and present structure, whereas Arabic has one tense that encompasses both of those English structures.

2.2.3 Types of Machine Translation

Due to these challenges it is clear that the creation of general purpose, Fully Automatic, High Quality, General-Purpose Machine Translation is still a distant goal. Therefore in this category there are two approaches [25]:

- Human Aided MT (HAMT)
- Machine Aided Human Translation (MATH) systems

2.2.3.1 Human Aided MT (HAMT)

In HAMT systems, the translator is a computer program. Pre and Post editing are done on the input and output texts respectively, to ease the analysis and generation processes [25].

Pre-editing: checking through the source text for foreseeable problem of the MT system and attempts to remove them.

Post editing: modifies the translation program's output to improve its readability and idiomatic.

2.2.3.2 Machine Aided Human Translation (MAHT)

In this case human beings ultimately do the translation. However, translators use various supports from the computer, like online dictionaries, Terminology Data Banks and Translation Memories (fragments of translations of previously translated texts).

2.2.4 Translation Architecture

There is three different approaches to MT have been used: direct translation; transfer based translation and interlingual translation. The current architectures of MT can be graphically represented as in (figure 2.1), which is a modification of the original Vauquois pyramid (figure 2.2). The vertical direction represents the amount of effort necessary for analysis and generation while the amount of effort needed for transfer increases with the width of the pyramid.

The base of the pyramid needs the most transfer and the least analysis and generation, while the top of the pyramid needs the least transfer and the most analysis and generation.

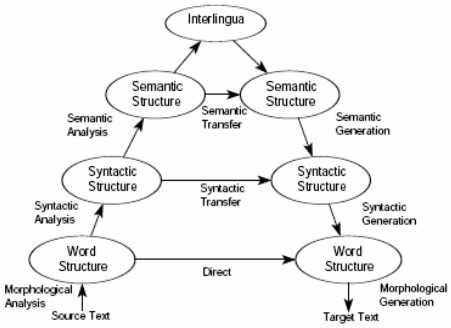


Figure 2.1: Levels of translation

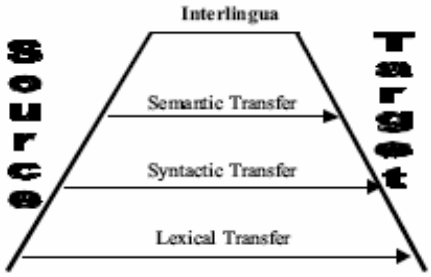


Figure 2.2: A Vauquois pyramid

2.2.4.1 Direct MT system

The direct method, also known as the transformer method, in which words in the source language are replaced with words in the target language (figures 2.3). In order to improve the output quality, some direct MT systems perform some morphological analysis before the bilingual; dictionary look-up but they rarely analyze the sentence structure of the source language text[17]. For example:

Bird fly→ Panchi udya; ਪੰਛੀ ਉਡਿਆ◌ਾ.

The basic characteristic for such type of translation is that it is very simple and one needs to replace a word of source language to a word in target language using a bilingual dictionary.

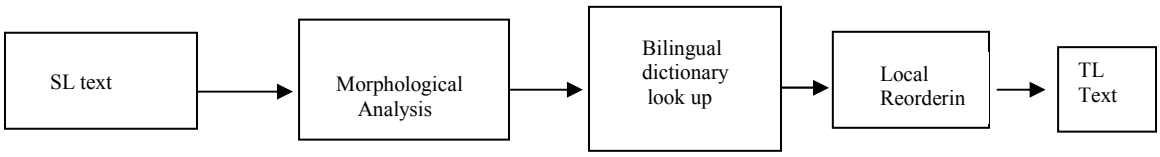


Figure 2.3 Direct machine translation system

The disadvantage of direct method is that it is unidirectional i.e. if the target is to be translated back into the source language, a different transformer must be used. It uses n^2 translation modules for translations among n languages, thus making it exponentially large for multi-language translating. Other problem with the direct method evolves if the structure of sentence is complex. It requires complex grammatical analysis, word ordering in the target language sentence can often be wrong. Additionally, if lexical ambiguity exists, incorrect translation of words occurs. Analysis of relations between different parts of the sentence is often lacking, which can lead to poor or unintelligible translations.

Direct translation is very inaccurate for complex texts, but has been implemented successfully for specialized corpora with a limited number of lexical entries.

Historical Developments

In 1950's the research under Erwin Reifler at the University of Washington used the dictionary-based 'direct' approach; it involved the construction of large bilingual dictionaries where lexicographic information was used not only for selecting lexical equivalents but also for solving grammatical problems without the use of syntactic analysis. After initial work on German and English, the group was engaged on Russian-English system.

The "first generation" research of the pre-ALPAC period (1956-1966) had been dominated by mainly 'direct translation' approaches, the "second generation" post-ALPAC was to be dominated by 'indirect' models, both interlingua and transfer based.

2.2.4.2 Transfer Based MT System

In order to overcome the major shortcomings of direct translation, researchers began working on the transfer method. It occupies the level above direct translation in the MT pyramid (figure 2.4 and figure 2.5) and is also known as indirect or linguistic knowledge (LK) translation. It requires linguistic knowledge of the source and target languages as well as the differences between them. The transfer architecture not only translates at the lexical level, like the direct architecture, but syntactically and sometimes semantically as shown in figure 2.4. The transfer method will first parse the sentence of the source language. It then applies rules that map the grammatical segments of the source sentence to a representation in the target language. The transfer method will first parse the sentence of the source language.

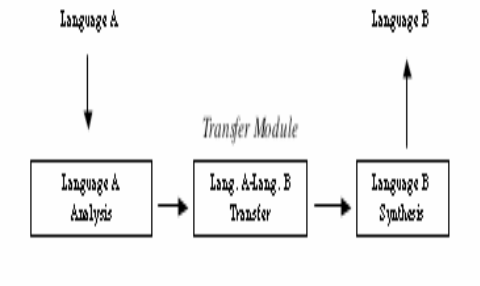


Figure 2.4: Transfer Based representation

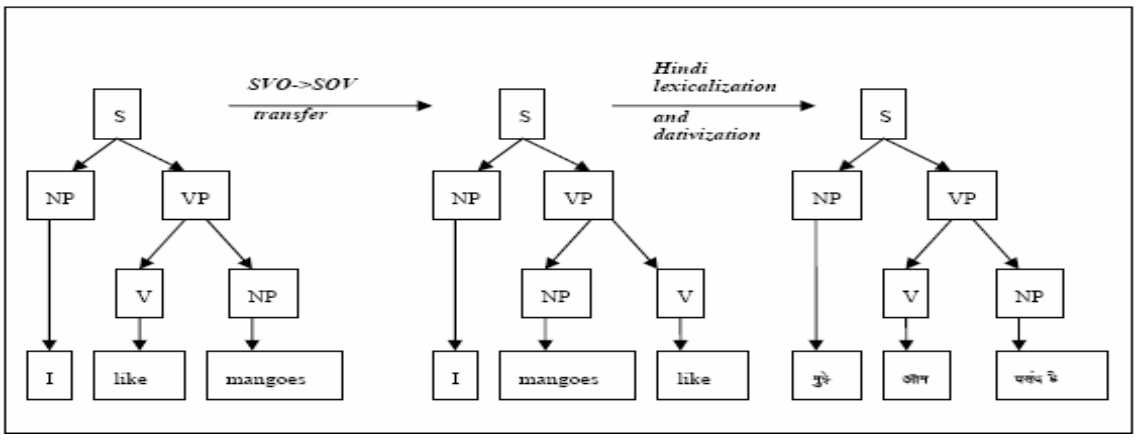


Figure 2.5: Translation of English Text to Hindi Text with Translation Approach

It then applies rules that map the grammatical segments of the source sentence to a representation in the target language

For Example:

Children Like Sweets → Bacche mithai pasand karde Han

Like-Verb Phrase Pasand Karde Han – Verb Phrase

Children- Subject(NP) Bachhe – Subject(NP)

Sweets – Object (NP) Mithai – Object(NP)

Above example shows English to Punjabi translation of a sentence. After syntactically and semantically analyzing the sentence, we can easily translate a sentence even with different structures like SVO → SOV.

The transfer approach uses n^2 transfer modules, *n* analysis components, and *n* synthesis components, where n is the number of languages in the translation system. Thus, one of its downfalls is the sheer size of the rules needed for its implementation.

Historical Developments

At Montreal, research began in 1970 on a syntactic transfer system for English-French translation. The TAUM project was under this direction for translation of weather forecast and developed by the Prolog programming language. It has been successfully operating since 1976. The TAUM group attempted to repeat this success with another sub language, that of aviation manuals, but failed to overcome the problems of complex noun and phrases, and the project ended in 1981.

Anglabharati, an Indian system developed by IIT Kanpur under the expert guidance of Dr R M K Sinha deals with machine translation from English to Indian languages, primarily Hindi, using a rule-based transfer approach.

MaTra is another India based Human-Assisted translation project for English to Indian languages based on a transfer approach.

The Computer Science Department at the University of Hyderabad has worked on an English-Kannada MT system, using the Universal Clause Structure Grammar (UCSG) formalism, also invented there. This is essentially a transfer-based approach, and has been applied to the domain of government circulars, and funded by the Karnataka government. The Jadavpur University at Kolkata has recently worked on a rule-based English-Hindi MAT for news sentences using the transfer approach.

2.2.4.3 Interlingua Based MT System

The Interlingua, or pivot, approach appears at the apex of the MT pyramid. The main idea behind it is that the analysis of any source language should result in a language-

independent representation. The target language is then generated from that language-neutral representation [18].

This approach requires only one interlingual transfer model whereas the transfer approach requires n^2 transfer modules. The interlingual approach, in theory, requires more analysis and is more abstract [15]. The interlingual approach requires *n* analysis components, one interlingua converter, and *n* generation components, where *n* is the number of languages in the translation system.

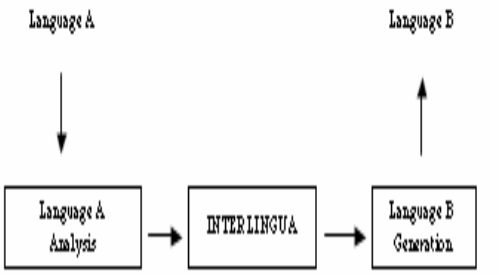


Figure 2.6: Interlingua representation

For Example:

Interlingual representation of sentence Dog barks (using UNL representation):

agt(*bark(icl>do),dog(icl>animal))*

There are a few problems with the interlingual approach. The interlingual approach requires an analyzer for each source language and a generator for each target language. Analysis of source text requires a deep semantic analysis that requires extensive world knowledge. Unfortunately, the true meaning of a sentence cannot always be extracted. Additionally, if a text is analyzed as deeply as is expected, then much of the source author's style will be lost [17].

Historical Developments

During 1950's the Soviet Union research was mainly focused on Interlingua approach. Between 1960 and 1971 Grenoble University developed a system for translating Russian mathematics and physics texts into French. It was not a pure interlingua as it did not provide interlingual representations for lexical items – these were translated by a bilingual transfer mechanism. A similar model was adopted at the University of Texas during the 1970s in its METAL system for German and English[16].

During the latter half of the 1980s [25], there was a general revival of interest in Interlingua systems, motivated by the contemporary research in artificial intelligence and in linguistics.

Another Interlingua project was also started in Netherlands. The important feature of this project was the exploration of the reversibility of grammars. This reversibility became a feature of many subsequent MT projects [17].

Finally, at the end of the 1990s, the Institute of Advanced Studies of the United Nations University (Tokyo) began its multinational interlingua based MT project – based on a 'standardised' intermediary language, UNL (Universal Networking Language). The Universal Networking Language (UNL) is an international project of the United Nations University, with an aim to create an Interlingua for all major human languages. It was initially for the six official languages of the United Nations and other widely spoken languages including Hindi (Arabic, Chinese, English, French, German, Indonesian, Italian, Japanese, Portuguese, Russian, and Spanish) – involving groups in some 15 countries.

2.2.5 Models of MT Research

Although there are only three main approaches to MT architecture (direct, transfer, interlingua), one may apply several informational components to increase the efficiency of the translating system. Such information components include: Knowledge-Based, Statistical and Example-Based.

2.2.5.1 Knowledge-Based MT

Knowledge-Based MT (KBMT) is characterized by a heavy emphasis on functionally complete understanding of the meaning of the source text prior to translation to the target text. KBMT does not require total understanding, but assumes an interpretation engine can achieve successful translation into several languages. KBMT is implemented on the interlingual architecture; it differs from other interlingual techniques by the depth to which it will analyze the source language and

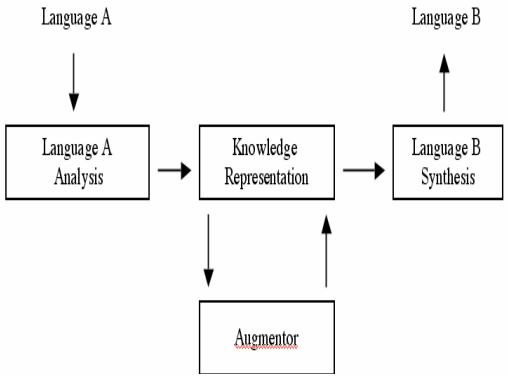


Figure 2.7: KBMT symbolic representation.

its reliance on explicit knowledge of the world. KBMT systems must be supported by world knowledge and by linguistic semantic knowledge about meanings of words and their combinations. Thus, a specific language is needed to represent the meanings of sentences. In many KBMT systems, frames that have named slots or features and values represent knowledge.

2.2.5.2 UNL Based MT System

The Universal Network Language (UNL) is an electronic language for computers to express and exchange every kind of information. UNL has been designed at the United Nations University (UNU)/Institute of Advanced Studies (IAS), Tokyo in 19990.

The motivation behind UNL is to develop an interligual representation such that semantically equivalent sentences of all languages have the same interlingual representation. It is an application of semantic net knowledge representation schema. It intermediates understanding among different natural languages. Thus, UNL is an intermediate language to be used through the Internet, which allows communication among people of different languages using their mother tongue. Information expressed in UNL can be converted into the native user's native language with higher quality and fewer mistakes than the computer translation systems. In addition UNL unlike natural language is free from ambiguities.

2.2.5.3 Example-Based MT

The example-based approach was founded on processes of extracting and selecting equivalent phrases or word groups from a databank of parallel bilingual texts, which have been aligned either by statistical methods or by more traditional rule-based methods. The main advantage of the approach (in comparison with rule-based approaches) is that since the texts have been extracted from databanks of actual translations produced by professional translators there is an assurance that the results will be accurate and idiomatic.

The idea behind *Example-Based* MT (EBMT) is to translate a sentence using previously analyzed examples of similar sentences to form proper representation. A database of previously analyzed text is stored in the *Translation Memory* (TMEM). TMEMM enables translators to store original texts and their translated versions side by side, i.e. so that corresponding sentences of the source and target are aligned. The translator can thus search for phrases or even full sentences in one language in the translation memory and have displayed corresponding phrases in the other language, either exact matches or approximations. Ideally, it will find an exact structural match for the source sentence and replace the example target words with the source target words. However, it is often the case that there is no exact match for a source sentence. In this case, the system will chunk the source sentence and try to find a match in the example database.

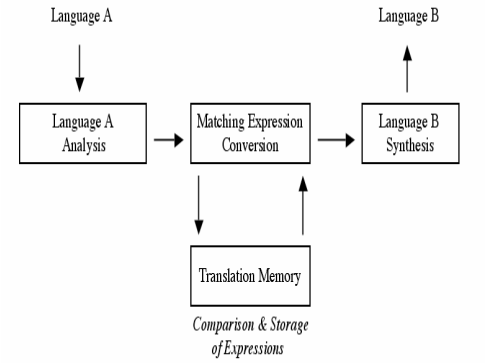


Figure2.8: Symbolic representation of EBMT

There are many approaches to storing translation pairs in TMEM. Text can be stored as complete parse tree, complete sentences, or phrases.

2.2.5.4 Statistical MT

It is a new method and its strategies are based statistics-based approaches. Now statistical methods were used as the means of analysis and generation; no linguistic rules were applied. The essence of the method was first to align phrases, word groups and individual words of the parallel texts, and then to calculate the probabilities that any one word in a sentence of one language corresponds to a word or words in the translated sentence with which it is aligned in the other language. What surprised most researchers were that the results were so acceptable: almost half the phrases translated either matched exactly the translations in the corpus, or expressed the same sense in slightly different words, or offered other equally legitimate translations. Thus, SBMT will pick the word (w) that has the highest probability of occupying its current position, given the surrounding words (s).

RAND Corporation undertook statistical analyses of a large corpus of Russian physics texts, to extract bilingual glossaries and grammatical information. The IBM India Research Lab at New Delhi has recently initiated work on statistical MT between English and Indian languages, building on IBM's existing work on statistical MT.

2.2.6 Machine Translation in India

Machine Translation in India is relatively young. The earliest efforts date from the late 80s and early 90s. The prominent among these are the projects at IIT Kanpur, University of Hyderabad, NCST Mumbai and CDAC Pune. The Technology Development in Indian Languages (TDIL), an initiative of the Department of IT, Ministry of Communications and Information Technology, Government of India, has played an instrumental role by funding these projects. Since the mid and late 90's, a few more projects have been initiated—at IIT Bombay, IIIT Hyderabad, AU-KBC Centre Chennai and Jadavpur University Kolkata[3].

There are also a couple of efforts from the private sector - from Super Infosoft Pvt Ltd, and more recently, the IBM India Research Lab.

Major MT Projects in India

They are some significant development in MT in India. Here are some important projects:

2.2.6.1 Anglabharati (and Anubharati)

Anglabharati deals with machine translation from English to Indian languages, primarily Hindi, using a rule-based transfer approach. It uses post-editing to resolve ambiguity/complexity. It is mainly developed for public health domain. The project is primarily based at IIT-Kanpur. Anubharati is a recent project at IIT Kanpur, dealing with template-based machine translation from Hindi to English, using a variation of example-based machine translation [11].

2.2.6.2 Anusaaraka

The focus in Anusaaraka is not mainly on machine translation, but on Language Access between Indian languages. Using principles of Paninian Grammar (PG), and exploiting

the close similarity of Indian languages, an Anusaaraka essentially maps local word groups between the source and target languages. The user needs some training to understand the output of the system. Its approach and lexicon is general, but the system has mainly been applied for children's stories. It was funded by TDIL[11].

2.2.6.3 UNL-based MT between English, Hindi and Marathi

The Universal Networking Language (UNL) is an international project of the United Nations University, with an aim to create an Interlingua for all major human languages. IIT Bombay is the Indian participant in UNL, and is working on MT systems between English, Hindi and Marathi using the UNL formalism. This essentially uses an interlingual approach—the source language is converted into UNL using a denconverter', and then converted into the target language using a 'deconverter' [10].

2.2.6.4 MaTra

MaTra is a Human-Assisted translation project for English to Indian languages, currently Hindi, essentially based on a transfer approach (Rao D, 2000). The MaTra lexicon and approach is general-purpose, but the system has been applied mainly in the domains of news, annual reports and technical phrases, and has been funded by TDIL.

2.2.6.5 Mantra

The Mantra project is based on the TAG formalism from University of Pennsylvania. A sub-language English-Hindi MT system has been developed for the domain of gazette notifications pertaining to government appointments. Recently, work has been initiated on other language pairs such as Hindi-English and Hindi-Bengali, as well as on extending to the domain of parliament proceeding summaries. TDIL and Department of Official Languages have funded the project.

2.2.6.6 UCSG-based English-Kannada MT

The CS Department at the Univ of Hyderabad has worked on an English-Kannada MT system, using the Universal Clause Structure Grammar (UCSG) formalism, also invented

there. This is essentially a transfer-based approach, and has been applied to the domain of government circulars, and funded by the Karnataka government.

2.2.6.7 Tamil-Hindi Anusaaraka and English-Tamil MT

The Anna University KB Chandrasekhar Research Centre at Chennai was established recently, and is active in the area of Tamil NLP. A Tamil-Hindi language accessor has been built using the Anusaaraka formalism described above. Recently, the group has begun work on an English-Tamil MT system.

2.2.6.8 English-Hindi MAT for news sentences

The Jadavpur University at Kolkata has recently worked on a rule-based English-Hindi MAT for news sentences using the transfer approach.

2.2.6.9 Anuvadak English-Hindi software

Super Infosoft Pvt Ltd is one of the very few private sector efforts in MT in India. They have been working on software called Anuvadak, which is a general-purpose English-Hindi translation tool that supports post-editing.

2.2.6.10 English-Hindi Statistical MT

The IBM India Research Lab at New Delhi has recently initiated work on statistical MT between English and Indian languages, building on IBM's existing work on statistical MT.

MT is relatively new in India – about a decade old. In comparison with MT efforts in Europe and Japan, which are at least 3 decades old, it would seem that Indian MT has a long way to go. However, this can also be an advantage, because Indian researchers can learn from the experience of their global counterparts.

Chapter 3: Universal Networking Language

The Universal Network Language (UNL) is an electronic language for computers to express and exchange every kind of information. UNL has been designed at the United Nations University (UNU)/Institute of Advanced Studies (IAS), Tokyo in 19990.

The motivation behind UNL is to develop an interligual representation such that semantically equivalent sentences of all languages have the same interlingual representation. It is an application of semantic net knowledge repsentation schema. It intermediates understanding among different natural languages. Thus, UNL is an intermediate language to be used through the Internet, which allows communication among people of different languages using their mother tongue [2].

Information expressed in UNL can be converted into the native user's native language with higher quality and fewer mistakes than the computer translation systems. In addition UNL unlike natural language is free from ambiguities [21].

3.1 UNL's Representation of Information

The UNL represents information sentence by sentence. Each sentence is converted into a directed hyper graph having concepts as nodes and relations as arcs[10].

The knowledge within document is expressed in three dimensions:

- Word knowledge is expressed by Universal Words (Uws).
- Concept Knowledge is captured by relating UWs through a set of UNL relations.
- Speakers view, aspect, time of event, etc. are captured by UNL attributes.

3.1.1 Universal Word

Universal Words (UWs) are character-strings used to represent simple or compound concepts It is made up of a character string followed by a list of constraints.

<UW>	::= <Head Word> [<Constraint List>]
<Head Word>	::= <character>…
<Constraint List>	::= "(" <Constraint> ["," <Constraint>]… ")"

Head Word

Head Word is an English word/compound word/phrase/sentence that is interpreted as a label for a set of concepts. UWs are used to index the UNL knowledge base (UNLKB).

For example: drink, eat, dog etc.

Constraints or Restrictions

The Constraint List restricts the range of the concept that a Basic UW represents. Each restricted UW represents a more specific concept, or subset of concepts.

For example:

state(equ>nation) :denotes nation.

state(icl>situation) : kind of situation

state(icl>government) :kind of government

3.1.2 Relations

Binary relations are the building blocks of UNL sentences. They are made up of a relation and two UWs. The relations between UWs in binary relations have different labels according to the different roles they play. There are many factors to be considered in choosing an inventory of relations[21].

For example:

Relation		**Description**
Agt	agent	a thing which initiates an action
Bas	Basis	a thing used as the basis
Con	condition	a non-focused event which conditioned a focused event
Dur	duration	a period of time during an event occurs or a state exists
Fmt	range	a range between two things
Icl	A kind of	a more general concept

man	manner	the way to carry out event or characteristics of a state
Nam	name	a name of a thing
Obj	affected thing	a thing in focus which is directly affected by an event or state
Opl	affected place	a place in focus where an event affects
qua	quantity	a quantity of a thing or unit
Rsn	reason	a reason that an event or a state happens

"Dog barks" is represented as:

agt (do, thing): agt(bark(icl>do), dog(icl>animal))

3.1.3 Attributes

Attributes of UWs are used to describe subjectivity of sentences. They show what is said from the speaker's point of view. Relations and UWs are used to describe the objectivity information of sentences. Attributes modify Uws to indicate subjectivity information such as about how the speaker views these states-of-affairs and his attitudes toward them and to indicate the property of the concepts[19].

For example:

Time with respect to the speaker	@past, @present, @future
Speaker's view of Aspect	@begin, @complete, @continue
Speaker's view of Reference	@generic, @def, @indef
Speaker's Focus	@emphasis, @entry, @qfocus
Speaker's attitudes	@confirmation, @exclamation, @interrogative
Speaker's view point	@ability, @although, @conclusion, @doubt
Describing speaker's attitudes	@polite, @request

"It began to work again" has attributes: work.@begin.@past

"I have done it" has attributes: do.@end.@present

3.1.4 UNL Knowledge Base

UNL Knowledge Base (UNLKB) defines every possible relation between concepts. The possible relations are defined based on a hierarchy of Uws. The UW System is built up by inclusive relations between concepts according to property inference mechanism of concepts. The architecture of the UW System allows introducing and defining any concept no matter how particular or specific it is. UNLKB not only provides linguistic knowledge in the form that computer can understand but also provides the semantic background of UNL expressions, that is the UNLKB ensures the meanings of UNL expression [10].

3.2 UNL Systems

Basically UNL system basically has the following components:

- Language Server
- UNL Editor and viewer
- UNL Proxy Server

3.2.1 Complete UNL System

Conceptually UNL works as follows[26]:

- It process input text and produces the UNL as output. A program module used for this process is called as Enconverter.
- Any one who wants to read this UNL document in his native language will use a module, which will produce equivalent natural text in his choice. A program module for this generation is called as Deconverter.

A conversion system from native languages into UNL is called "enconverter", and the one that deconverts from UNL into native languages is called "deconverter. Information "enconverted," from any language is exchanged in UNL format via networks. Information represented in UNL is "deconverted" into each native language on the terminal network. The processes of “enconversion” and “deconversion” are provided

by a Language Server, which resides in the network of the Internet. The "enconverter" and "deconverter" are responsible for converting a particular language into UNL, and vice versa. The "enconverter" enconverts a language into UNL, while the "deconverter" deconverts UNL into a native language[24].

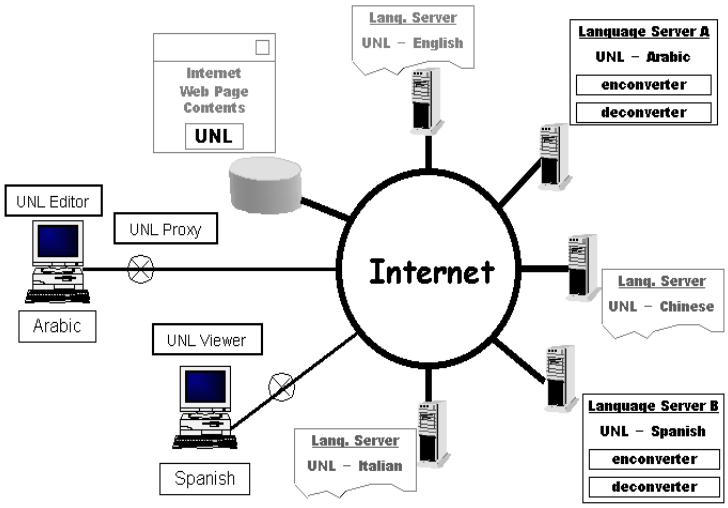


Figure 3.1 UNL Systems

.

In this example, the Arabic Language Server and the English Language Server provide the conversion service. When home pages are developed in Arabic, the UNL Editor recognizes the contents as Arabic and sends a request to the Arabic Language Server to "enconvert" the text. Once the Arabic text is "enconverted" to UNL, the Arabic Language Server sends the results back to the UNL Editor. Home page designers can now embed UNL into their pages. When we read this page in English, the UNL Viewer recognizes the contents as UNL and sends a request to the English Language Server to "deconvert" the text. Once UNL is "deconverted" to English, the English Language Server sends the results back to the UNL Editor. Hence, the text – once converted to UNL – may be deconverted to many different languages.

3.2.1.1 Language Server

Language server consists of deconverter and enconverter. The processes of "enconversion" and "deconversion" are provided by a Language Server which resides in the network of the Internet.

The "enconverter" and "deconverter" are responsible for converting a particular language into UNL, and vice versa. The "enconverter" "enconverts" a language into UNL, while the "deconverter" "deconverts" UNL into a native language[23].

3.2.1.1.1 Enconverter

An "enconverter" is a software that automatically or interactively enconverts natural languages text into UNL. UNU/IAS developed a software for enconversion called "EnCo" which constitutes an enconverter together with a word dictionary, co-occurrence dictionary and conversion rules for a language. This "EnCo" is a language independent software, then it is applicable for any languages.

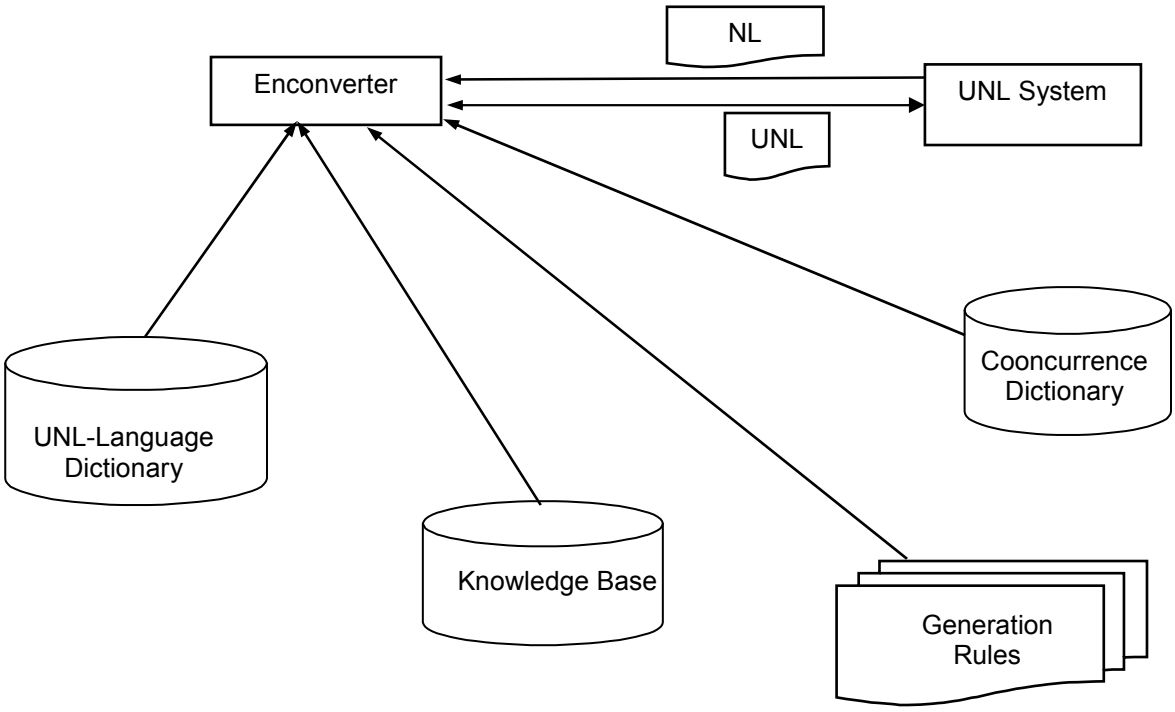


Figure 3.2 Enconverter

An "enconverter", as it generates UNL from natural languages, enables people to make UNL documents without any knowledge about UNL. It means that users of the UNL system do not need learn UNL. This makes UNL quite different from Esperanto, for instance.

3.2.1.1.2 Deconverter

A "deconverter" is software that automatically deconverts UNL into native languages. It is important to achieve a high quality and correct results. It is also important that the basic architecture of the "deconverter" is widely shared throughout the world, in order to treat all languages with the same quality and precision standards. Technology developed for a language can be applied to other languages as long as the architecture is shared. A "Deconverter", which generates natural language from UNL, plays a core role in the UNL system .It is very significant that "deconverter" is capable of expressing UNL information with very high accuracy.

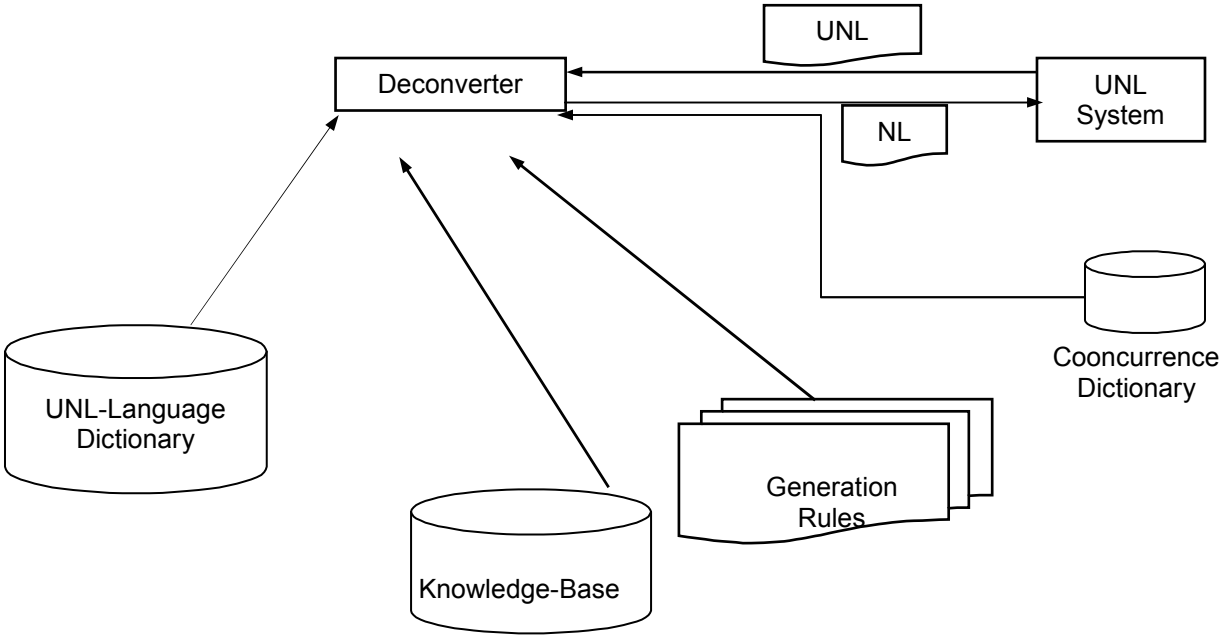


Figure 3.3 Deconverter

It follows that information, once composed in UNL, can be understood in any language as far as there be a "deconverter" of the language

3.2.1.2 UNL Editor and Viewer

UNL editor is used to make UNL documents. UNL editor is linked to language server equipped with a "enconverter" and a "deconverter" for a natural language. As the author writes a document, e-mail or any other text, in his/her language, UNL editor "enconverts" it into UNL documents. In this process, UNL expressions are produced automatically or interactively with the author.

One of the HTML merits is that it allows production of the whole document in plain text. In general, information contained in an electronic document is divided into text and embedded instruction. In HTML, however, even embedded instruction is also described in plain text. This characteristic gives HTML a universal adaptability to any editing system in holding the advantage of hypertext. Furthermore, in HTML, description format for embedding is open to the public. HTML conventions are still expanding and developing. Conventions to treat UNL information are expected to be regarded as one of extensions in HTML[22].

3.2.2 UNL Proxy Server

The UNL Proxy Server is a stand-alone application developed by java programming language and works in a terminal computer. It works as a filter that allows Internet browsers to recognize web pages written in UNL and engages the appropriate Language Server on the Internet so that the document can be read in a natural language.

Working

The user must adjust his or her browser settings to use the UNL Proxy Server. He or she then starts the Proxy server and sets it to access a desired Language Server. In the process of accessing a web page, the browser will pass the URL through

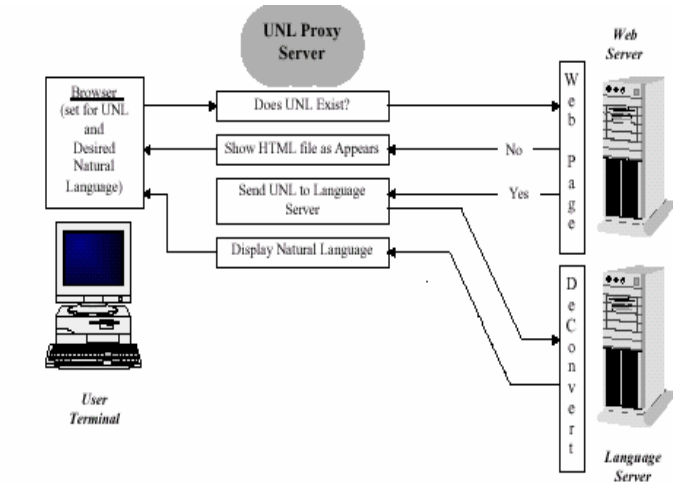


Figure3.4: UNL Proxy Server

the UNL Proxy. The UNL Proxy determines whether the file has a UNL extension and contains UNL expressions. If this is confirmed, the Proxy communicates with the appropriate Language Server specified by the user. The Language Server then "deconverts" the UNL expressions into the desired natural language, the result of which is sent back by the UNL Proxy to the browser. In case only HTML files are found, the Proxy allows them to be read as they appear. In this manner, if a user adjusts his or her browser settings to use the UNL Proxy server and selects Punjabi language for the UNL Proxy server, he or she will then view a web page in Punjabi, which is actually written in UNL.

The UNL system allows people to communicate with peoples of different languages in their mother tongue. The UNL is a common language to exchange information through computers, which can deal with natural languages.

Chapter 4: Punjabi Generator System

4.1 Problem Statement

A "Deconverter", which generates Punjabi language from UNL, plays a core role in the UNL system. It is very significant that "deconverter" will be capable of expressing UNL information with very high accuracy. It will consist of word dictionary and conversion rules for a language. This will be language independent software that is applicable for any languages. This engine takes UNL expression as input and generates target language (Punjabi) sentence with the help of various database files like lexicon files, morphological rule files [22].

The deconverter can be logically portioned into three phases as:

1) Syntax planning phase
2) Case marking phase
3) Morphology phase

The syntax planning phase is aimed at generation of proper sequence of words for the target sentence. These phases first reads the input UNL file and convert it into semantic-net like structure known as nodenet. Nodenet is a directed acyclic graph structure (DAG), which defines the sentence in the form of DAG. We use lexicon files to map the UWs to target language worlds. After generating a nodenet, the problem of the syntax plan generation get reduce to the problem of DAG traversal. Proper traversal of the node net generates the syntax plan of the target sentence.

The syntax planning phase generates the sequence of words, which cannot express the complete contents of the sentence. This syntax plan needs to be processed by the case-marking file, which apply proper case marker for each and every relations. This case-

marking phase is next processed by the morphology phase. The morphology phase gives a final form of the target sentence.

Here we discuss the structure of Punjabi sentence and based on this we describe the basic idea of the generator system.

4.2 Punjabi Sentence Structure and Representation

One of the main motivations towards the generation system is the subject-object-verb structure of Punjabi as against subject-verb-object structure of English [15].

Eg: Ram saw Sita *Ram ne Sita nu vekha*

Subject-verb-object Subject-Object-Verb

Punjabi is fairly flexible in placing of its subordinate clauses. The flexibility can be observed in this example. Consider the sentence:

"Ram saw Sita who lives in Delhi"

In Punjabi the sentence reads as:

Ram ne Sita nu vekha, jehri dilli wich rehndi hai

Ram ne Sita, jehri dilli wich rehndi hai, nu vekha

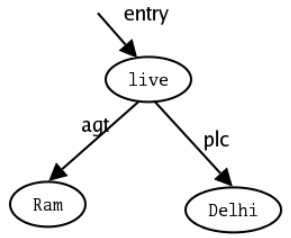
Jehri dilli wich rehndi hai ,Ram ne us Sita nu vekha.

All the three forms showed above are acceptable Punjabi sentences.

4.2.1 Simple UNL Representations

Simple sentences in UNL are represented through nodenet by taking into consideration all the relation it has in the UNL expression.

Consider the example: *Ram live in Delhi*



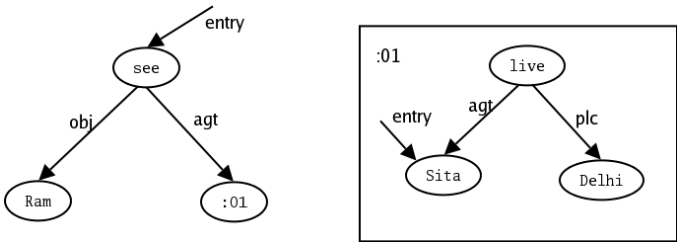
4.2.2 Multiple UNL Representations

Clausal sentences can be represented in more than one ways in UNL viz. either using a scope node (Compound-UW), or using multiple parents on some of the nodes.

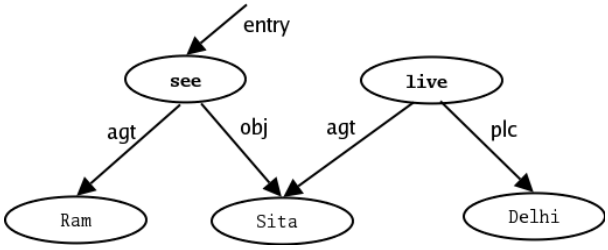
Consider the same example:

"Ram saw Sita who lives in Delhi".

This is the first representation: It is called the hyper-graph representation:

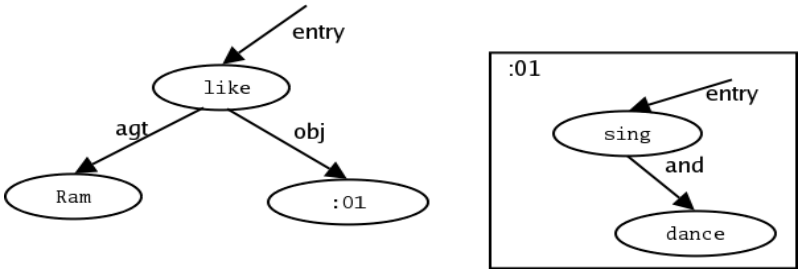


The second way to represent this sentence is using multiple parents:

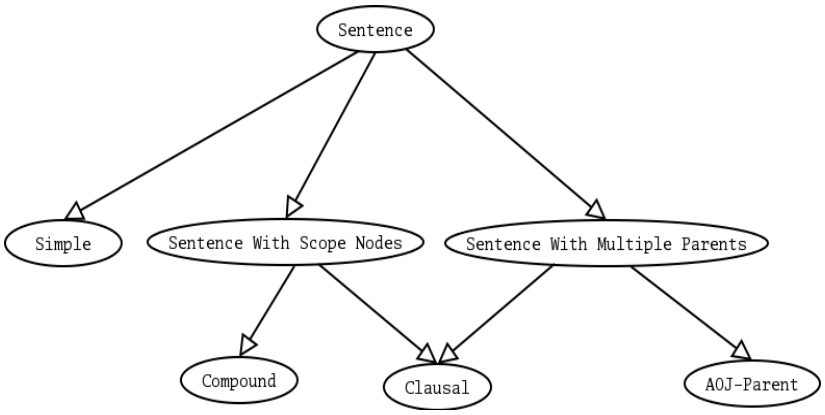


Both are valid UNL representations for the same sentence. In the representation, the node "live" has no parents. And starting from the node marked "entry", we cannot reach this node (live) by following only child pointers. Such nodes are called "Orphans"

However this is not the case with Compound Sentences. For example "Ram likes singing and dancing". It has only one representation

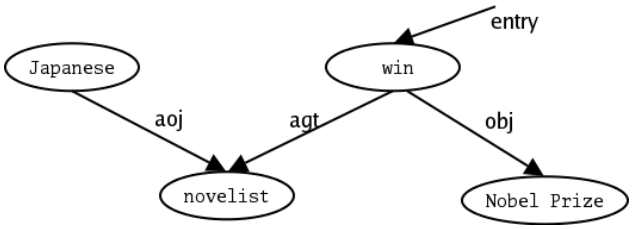


Based on this observation the *UNL representation* of a sentence can be classified in the following way:



4.2.3 The Case of aoj-Parents:

Multiple parents can also be encountered in case of aoj-relation like in the following example: "Japanese novelist won Noble Prize". Its UNL-representation is:



Here even if we don't have a clausal or compound sentence we see a case of multiple-parent. This is a sort of exceptional behavior and is not shown by any other relation.

4.3 Steps for building a deconverter

The complete Generator System theoretically identifies three main phases in its process of converting UNL document into Punjabi text[7][9]:

1) **Syntax Planning**: In this phase we generate the order in which the Punjabi lexicons will appear in the final order. The order the Punjabi words are decided on the basis of relation labels only.
2) **Case Marker Generation for Individual Nodes**: In this phase we take care of what Case Marker should precede or follow the individual words. This decision is mainly taken on basis of relation labels and the attributes of the universal words.
3) **Morphology:** In this phase we take care of what kind of morphological transformations a Punjabi word should undergo.

The over all working system involves the following major steps[11]

Step 1:

The UNL document is parse for building the semantic Node-net. This Semantic node Net is representative of the meaning of the sentence. This node net by its very structure is a DAG (Directed Acyclic Graph).

Step 2:

Corresponding to each universal word parsed a reference to a Master Punjabi dictionary is made to pick up all the lexical information (gender, tense etc) about that particular Universal-Word. Punjabi word corresponding to the universal is also extracted in this step. This step is done while parsing[8].

Step 3:

Corresponding to each relation a reference is made to a Case Marker database. Then using the relation label, and lexical information available for the Universal-Word involved in that relation Case Markers are assigned to the nodes. This step is done while parsing.

Step 4:

A call to the morphology database is made and corresponding morphological transformations are done on each node. This step is also done while parsing.

Step 5:

After parsing is complete, the complete node net is generated with correct case markers and morphological forms for each Universal-Word; we do the syntax planning and generate the final output.

4.4 Syntax Planning

The syntax planning [1] phase is aimed to generate the proper syntax plan of the target sentence. Ideally, the syntax planning phase is supposed to position each and every word in the target sentence, in its proper position. To achieve this, the syntax planning phase is subdivided into the pipeline of following tasks:

- UNL Parser
- UW Resolution
- Node net Building
- Heuristics for Syntax Planning
- Traversing of the Nodenet

To achieve modularity in the design, internal data of the syntax-planning phase is conceptually divided into five separate groups. Different tasks of the syntax-planning phase communicate with these blocks. These blocks are as follows:

- **UW Repository:** It is used to store all the UWs present in the given UNL sentence. Following are the responsibilities of the UW repository:
  - ✓ It instantiates simple UWs and compound UWs as per the request.
  - ✓ It ensures unique instance of each UW.
- **Relation Repository:** This block is designed to store all the binary relations present in the given UNL sentence. Its responsibilities are:
  - ✓ To keep information related to every binary relation present in input UNL sentence.

- ✓ To construct the Nodenet structure defined by all the binary relations in the input sentence

- **Nodenet Block:** This block maintains critical information about the Nodenet structure. Following are the responsibilities of this block:
  - ✓ It keeps track of entry point to the Nodenet.
  - ✓ It executes various algorithms to get desired syntax plan from Nodenet structure.

The pipeline of the task, along the different logical data blocks in the system, is shown in the figure 4.1

4.4.1 Parsing of an Input UNL file

Parser is the important part of any conversion system. This system needs a parser to read an UNL file and convert it into machine understandable format. The syntax for an UNL file is as follows:

<UNL File> :: = [<Comment Lines>]<UNL Exprn>…..[<Comment Lines>]

<Comment Lines> :: = <Comment Line>….

<Comment Line> :: = “;”<Any Character Except Newline>”\n”

<UNL Sentence> :: = <UNL Sentence1> | <UNL sentence2>

<UNL Sentence1> :: = “[“{S|s}”]” <UNL Exprn>..”[\”{S|s}”]”

<UNL Sentence2> :: = “{un1}”<UNL Exprn>…”{\unl}”

<UNL Exprn> :: = <Relation Label><Scope ID>”(“<UW>”,”<UW>”)”

<Scope ID> :: = {<Digit>|<Character>}{<Digit>|<Character>}

<UW> :: = <Head Word> [<Constraint List>]

The parser implemented with the system parses the input UNL file, and performs the following tasks:

1. It adds simple and compound UWs to the UW repository.
2. It instantiates the relations and adds it to the relation repository.
3. It reports certain errors, if exists, in the input UNL file.

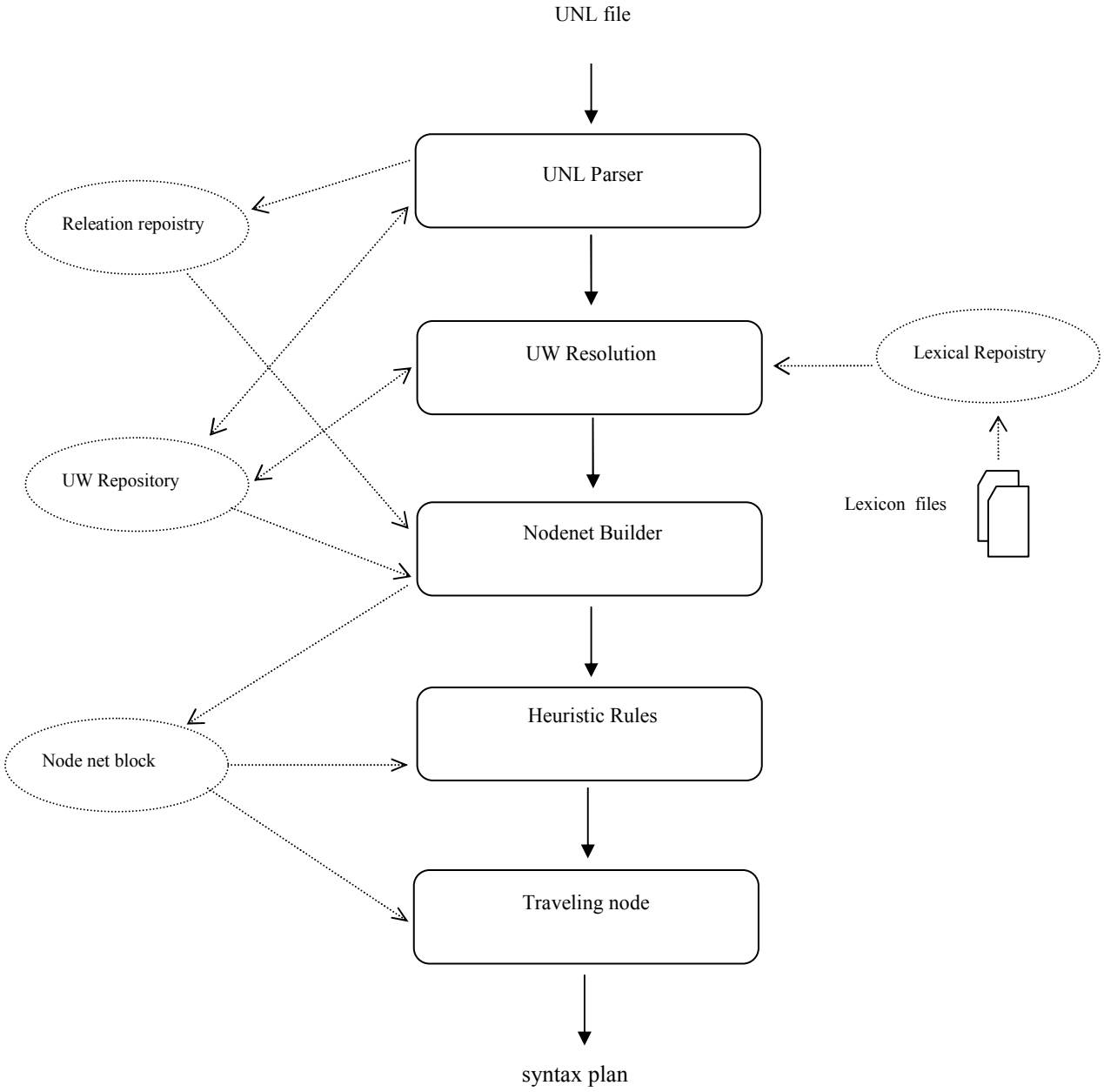


Figure 4.1 :Logical data blocks and subtasks of syntax planning phase

4.4.2. UW Resolution

Universal words (UWs) are basically tokens that stand for a particular concept. These UWs must be resolved into a equivalent target language words, with the help of lexicon. This generally involves contextual and pragmatic processing along with the use of co-occurrence dictionary for selecting appropriate target language word, when the UW under consideration has more than one entry in the lexicon. After selecting appropriate word, we store the target language word and its lexical information for further use. This lexical information can be referred to in the morphology phase and also helps in the positioning of words.

4.4.3 Building the Nodenet

The Nodenet is a Directed Acyclic Graph (DAG) structure [13], which graphically represent UNL sentence. A node in the Nodenet represents a concept (i.e., it stands for UW). An edge in the Nodenet represents relation between two nodes (i.e. it stands for a binary relation)[1]. All the edges are directed from the parent to child, i.e., from UW_1 to UW_2 in the expression rel (UW1, UW2). The edges are labeled with the corresponding relation label, as shown in the figure 4.2(b). The UW having attribute @entry forms the root of the Nodenet. In the Nodenet traversal, the traversal may encounter the nodes that have two or more parents.

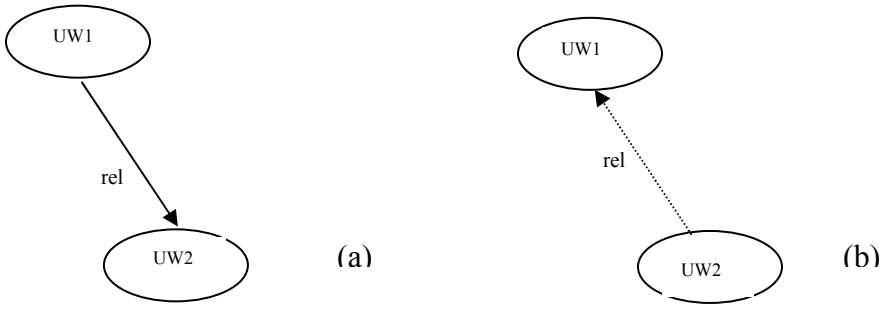


Figure 4.2 representation of binary releation

For example, in the Nodenet shown in the figure 4.3, traversal needs to visit node D from the node B. In such cases, to visit all possible parents, a mechanism is needed to access all parents of node under consideration. To provide such an access, backedges are kept in every child node. A back edge points to the parent of the node. For every parent of the node, a separate back edge is kept in node. All back edges are directed from child to the parent, i.e., from UW2 to UW1 in expression rel(UW1, UW2). Similar two edges, we also label back edges with relation label, as shown in figure 4.2 (b).

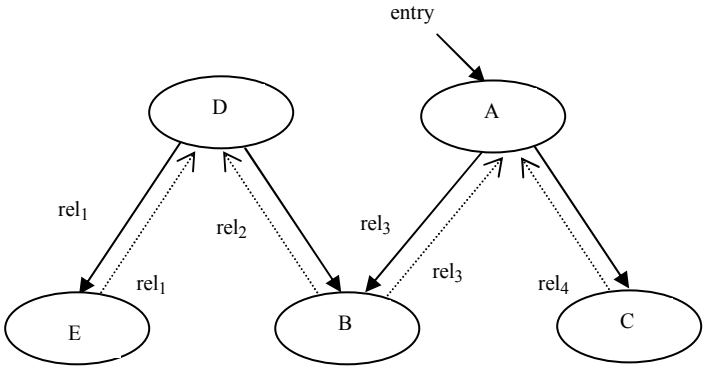
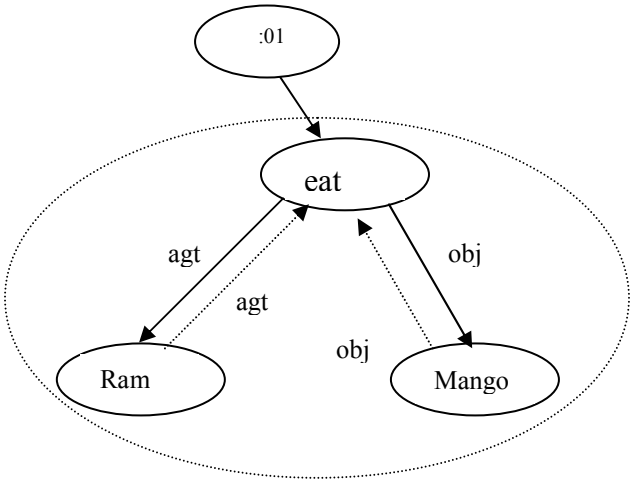


Figure 4.3 Example to illustrate need of a back-edge

More often, the UNL expression for complex sentences use compound UWs to represent various compound concepts in sentence. To represent a compound UW, in Nodenet, we use a scope node. A scope node is a node, which stands for a compound concept in the nodenet corresponding to the compound UW. Figure 4.4 shows snapshot of the Nodenet representation for a compound UW.



agt:01(eat@entry,ram)
obj:01(eat@entry,mango)

Figure 4.4:nodenet representation for compound UW

After resolving the UWs, the system is now ready with all required information to build nodenet. The pseudo code for building a nodenet is as follows:

Pseudo code for Nodenet building

```
for(every simple UW in UW repository)
        Create a simple node representing the simple UW;
for(every compound UW in the UW repository)
        Create a scope node representing compound UW;
for(every relation in relation repository)
{

        Node1 = Node representing UW1;
        Node2 = Node representing UW2;
        If ( relation is under the scope)
        Then
        {
                if(UW1 contains @ entry attribute)
                then
                        set the entry point of the corresponding scope node to point node1;
                else if(UW2 contains @ entry attribute)
                then
                        set the entry point of the corresponding scope node to point node2;
        }
        else

{
if (UW1 contains @entry attribute}
then
set the entry point of the Nodenet to point Node1;
else if(UW2 contains @entry attribute)
```

then

set the entry point of the *Nodenet* to point $Node_2$;

}

add an *edge* in the $Node_1$ pointing to the $Node_2$;

add a *back edge* in the $Node_2$ pointing to the $Node_1$;

}

In the above algorithm, we find the entry node and then make a edge to other node with which it is having releation and also make a back edge from it to the entry node. The above code transforms all the relations from relation repository into a Nodenet. Figure 4.5 shows the Nodenet, along with the important data-structures, for representing the following UNL expressions for the sentence *Ram who saved Shyam loves Sita.*

[S]

agt(love.@entry,:01)

obj(love.@entry,Sita)

agt:01(save,Ram.@entry)

obj:01(save,Shyam)

[\S]

rel	scope	UW1	UW2
agt	0	0	1
obj	0	0	5
obj	1	2	4
agt	1	2	3

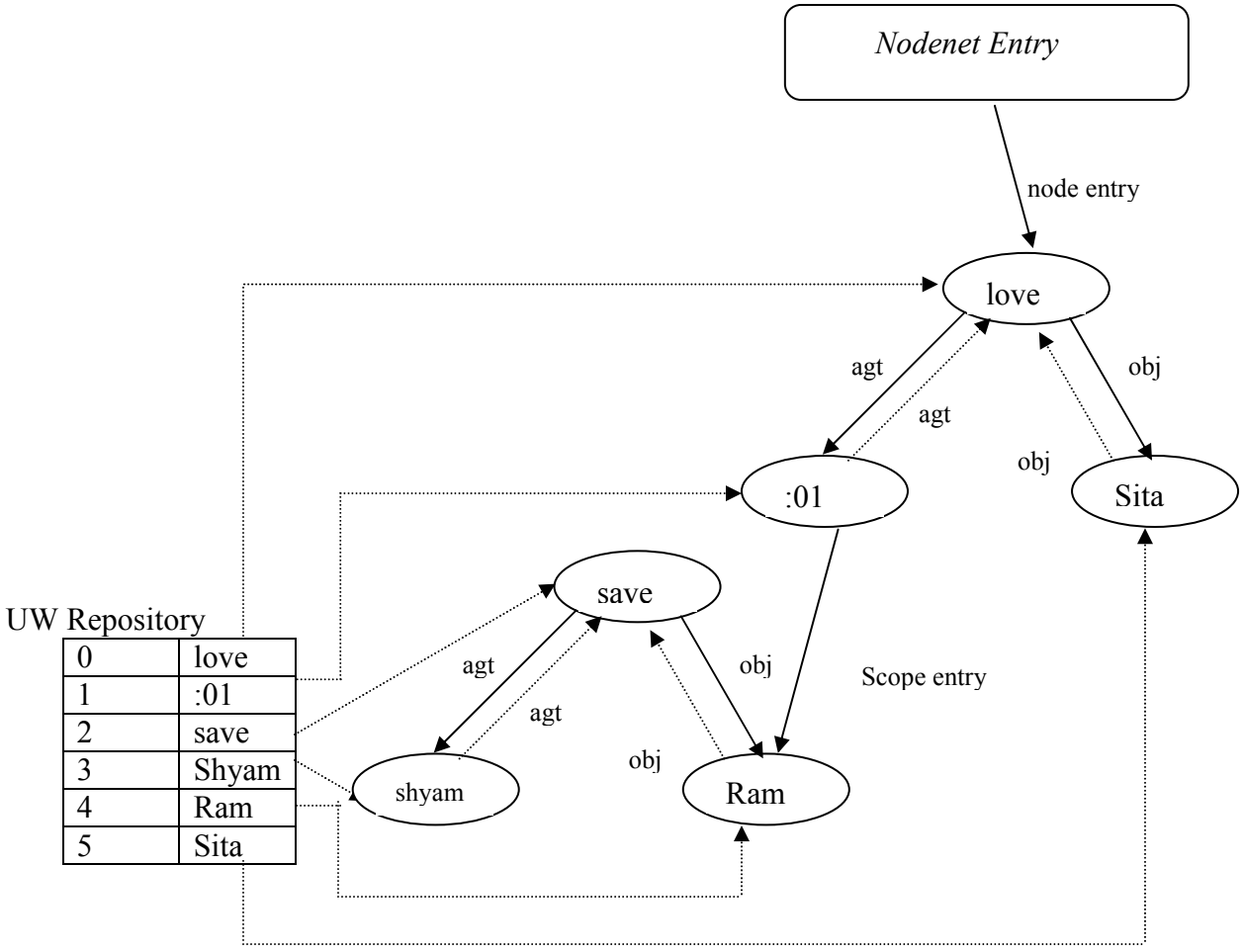


Figure4.5: organization of internal data structure

4.4.4. Heuristics for Syntax Planning

Heuristic for simple and clausal senences are explained separately below:

4.4.4.1 Heuristics for Syntax Planning of simple sentences

The main idea for syntax planning of simple sentence is a call to a recursive routine Syntax-Plan. This routine outputs the correct syntax plan for the sub-tree of the node on which it is called.

Its recursive nature can be explained in the following four elementary steps:

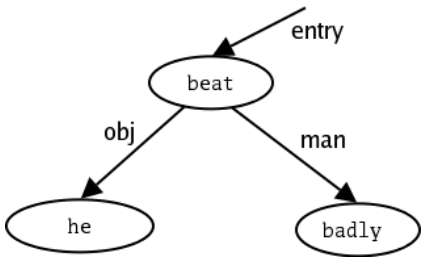
1) While on a given node *N* we identify all of its children. Say we have *n* children.
2) For each of these *n* children we obtain a correct order of their subtrees by calling Syntax Plan on each of these nodes. Now we have *n* phrases.
3) We now rearrange these n phrases and the node *N* among themselves based on heuristics, which are developed by expert linguists with their insight of target language during training phase of this tool.
4) Now we combine these *n* phrases and node *N* in the resultant order. This is the correct word order for the subtree of node *N*[15].

So to get the Syntax Plan of the complete sentence we call this routine on the entry node.

In this section we describe the heuristics that we came up with:

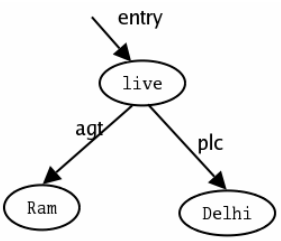
- If there is no child of the root node with "con" and "agt" then "obj" comes leftmost:(root node is any node under consideration)

  For example: He was beaten badly.



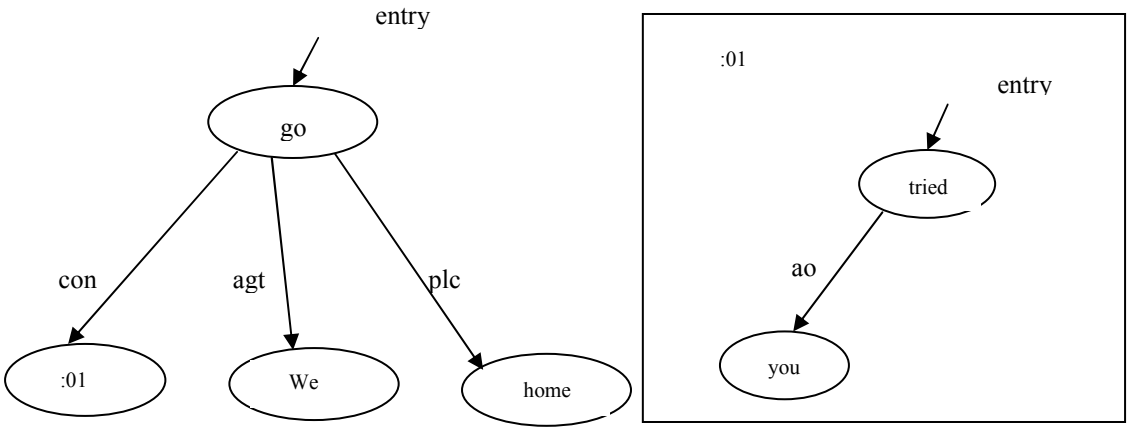
Uno buri tarah mara gaya.

- If there is no "con" and there is "agt" then "agt" is left most.
  For example: Ram lives in Delhi



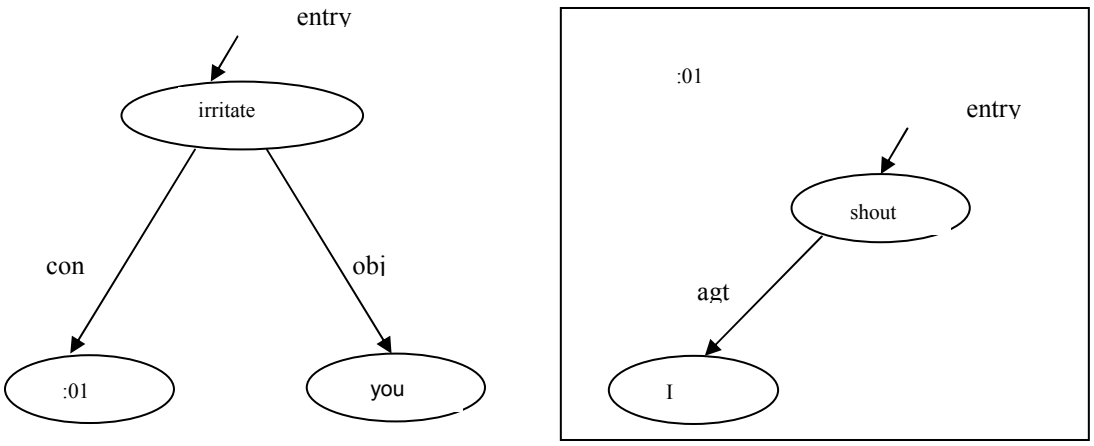
Ram dilli wich rehnda hai.

- If there is "con" then "agt" follows "con".
  For example: If you are tired, we will go straight home



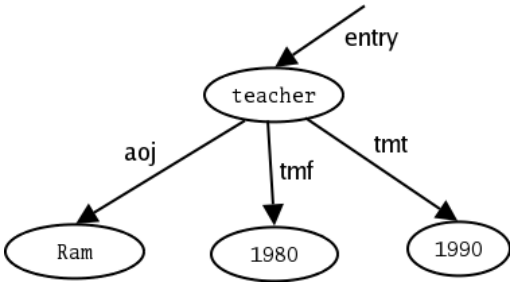
je tusi thak gaye ho , taan aapen gharen chalne haan

- If there is "con" and no "agt" then "obj" follows "con"
  For example If I shout, you will be irritated.



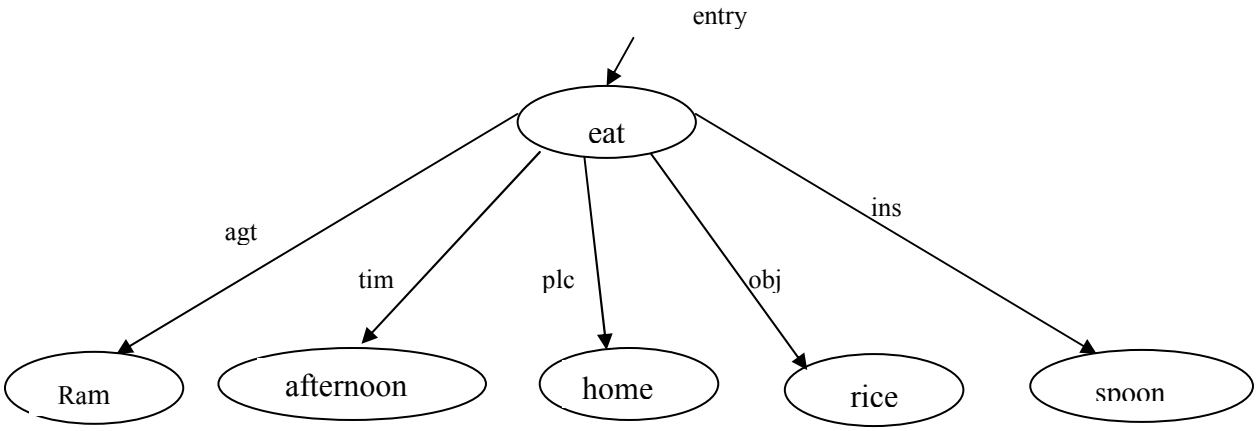
jeee mein shoor karan gaya ,teen tusi chird jayon gaye

If none of "con", "agt" and "obj" present then "aoj" comes left-most
For example: Ram was a teacher from 1980 to 1990.



Ram 1980 ton 1990 tak master see.

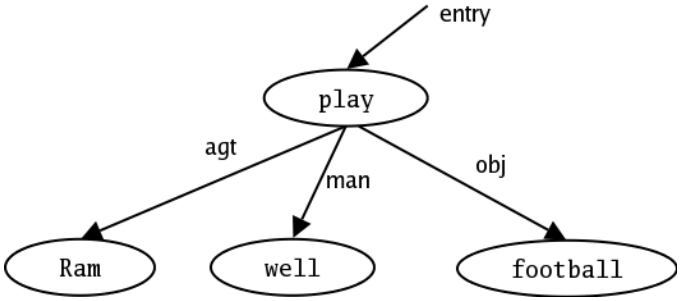
- If "agt" is present then "obj" goes right most [*see figure below*]
- If "agt" and "obj" are present then "ins" goes to right, just after object. [*see figure below*]
- If both "tim" and "plc" are there then they be aywhere but as always "tim" immedeately preceeds "plc"
  Eg: Ram eats rice in afternoon at his home with spoon



Ram dopehar wich chammache naal khana khanda hai.

- If "agt" present then "man" goes right after agt

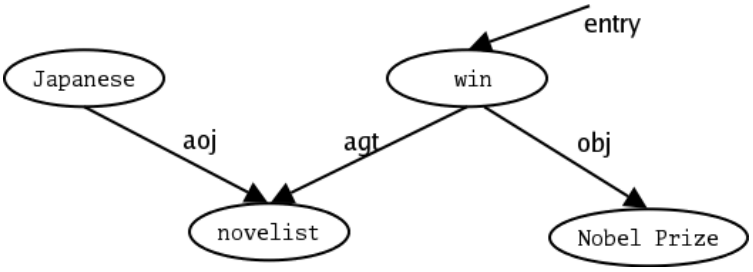
  For example : Ram plays football well.



Ram changa football khelda hai.

- If the root node has an "aoj" parent and if its not visited so far while graph traversal, then "aoj" parent precedes root word.

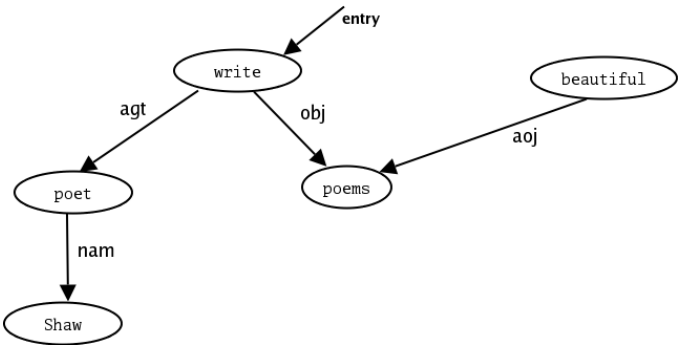
  For example:Japanese novelist won Nobel Prize.



Japani sahityakar ne Nobel Puraskar jeete.

- If root node has a "nam" child then it comes after root node.

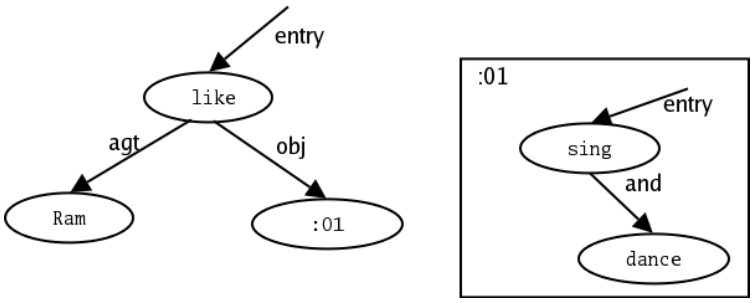
  For example :Poet Shaw wrote beautifull poems.



Kavi Shaw ne sohni kavitaen likhin.

- "and" child is the right most after root node.

  For example: Ram likes to sing and dance.



Based on these heuristics the decision about word order is taken. After this the Punjabi words are written in the out put file. And this completes the Syntax Planning Phase for simple sentences[15].summary of which is given below

Presence of relation	**Absence of releation**	**Order of relation**
obj	agt ,con	Leftmost
agt	con	Leftmost
agt	-	leftmost
agt, con	-	con agt
con obj	agt	con obj
aoj	Con agt obj	Leftmost
tim,plc	-	tim,plc
agt, obj	-	Obj is rightmost
agt,obj,ins	-	agt, obj ,ins
agt, man	-	Agt ,man
nam	-	After rootnode
and	-	After rootnode

4.4.4.2 Syntax Planning of Compound and Complex Sentences

As we have seen earlier that clausal sentences can be represented using two ways.:

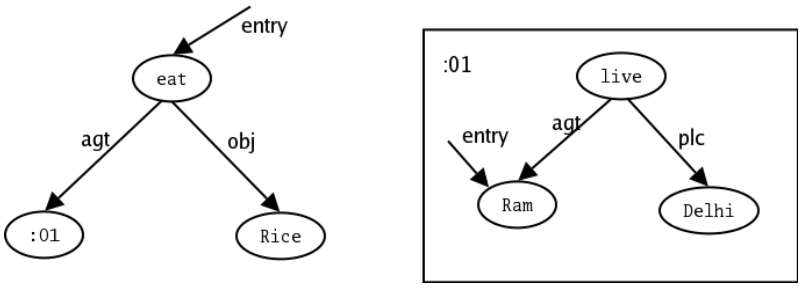
1) Using Scope Nodes
2) Using multiple parents.

So we need to tackle these two cases in our basic syntax plan system. We do it step by step.

Handling Scope Nodes:

We put the clause in the scope within the main clause. This is achieved by substituting the clause represented by the scope node at the position where the scope node appears in simple syntax planning [15]. Whenever we see a scope node, like other nodes give it a place in final order.

Firstly, we do the simple syntax planning on the clausal sentence. This will give us a syntax plan with compound uid between the other nodes. This compound uid represent the one of the scope node. Now we do the, syntax planning for the corresponding scope node and then replace the compound uid with corresponding syntax plan. For example, Ram who lives in Delhi eats rice.



In the first pass we generate:

:01 Rice eat

In the second pass we expand the Compound-Uid in their place.

[Ram Delhi Live] Rice eat

And this order is the final order that we want.

Ram Delhi Live Rice eat.

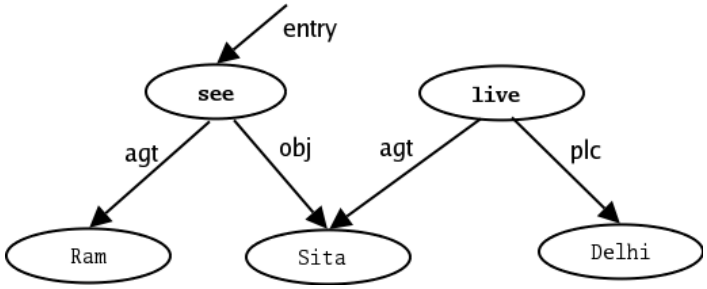
(Ram ,jehra dilli wich rehnda hai, chawal khanda hai.)

Handling Multiple Parent:

We add one more heuristics:

While on a given node N during traversal, if N has multiple-parents and if one the parents is having "V" attribute (off course this parent should not be the one which we used to reach N), then we do the Syntax_Plan on this other parent and place the generated clause right after N. By grammatical definition A clause is identified by presence of a verb. So this is what we look for in this heuristic.

For example: Ram saw Sita who lives in Delhi



The order from traversal for simple sentences that will be generated by a call on entry node will be:

Ram Sita see

Now when visiting node “Sita” we see a case with multiple parent and the other parent bearing an “agt” relation. So right after “Sita” we traverse node “live” and then leave node “Sita”

This would lead to the following order:

Ram Sita Delhi live see.

(Ram ne Sita nu, jehri dilli wich rehndi hai, vekha.)

Further when we are done with traversal of the entry node in any scope we check if there are any Orphan-nodes in that scope that have yet not been visited. Such nodes also

represent clauses and we do Syntax_Plan for these nodes now and put it in the end of the main clause of that scope.

4.4.4 Traversing the Nodenet

Final step of the generation of the syntax plan involves traversing the Nodenet according to the heuristic. The traversal algorithm visits nodes as per the demand of the heuristic of different relations related to the node. Traversal gives the exact syntax plan of the target sentence[1].

Thus at the end of the first step, we get the syntax plan i.e. exact order of the words for the given UNL file.

4.5 Case Marking

The syntax-planning phase is aimed at generation of proper sequence of words for words, but they cannot express the complete contents of the sentence. This syntax plan needs to be processed by the case marker phase. Case marker phase apply proper case marker for each and every releation in the given UNL expression i.e. it take into consideration Realtional Morphology.We follow a rule base approach to incorporate the case markers correctly [15]. A Case Marker data file contains one or more set of constraints for each relation and each of these sets map to different case markers. So given a node with all its attributes including lexical attributes from dictionary, we search the database for appropriate rule which the node satisfies and accordingly the case markers are initialized for the case markers [14].

4.5.1 Case Marker Database File:

Each line of the Case Marker Database file has 9 columns each separated by a conlon (‘**:**’) character. Every time a new relation is read from UNL document, this file if referenced once. The 9 fields are described as follows:

1. Relation Name – This is the name of the relation corresponding to which the reference to this file is being made.

2. Case Marker preceding Parent: If there is any case marker that will precede the Parent node in the relation, then that is stored in this field.for e.g.

   con: ਜੇnull: ਜੇ:null:null:null:null:null

3. Case Marker following Parent: If there is any case marker that will follow the parent node in the relation, then that is stored in this field. For e.g. and:null:ਤੋਂ:null:null:null:null:null:null. ਰਾਮ **ਤੇ** ਸ਼ਾਮ ਦੋਸਤ ਹਨ

4. Case Marker preceding Child: If there is any case marker that will precede the child node in the relation, then that is stored in this field.

5. Case Marker following Child: If there is any case marker that will follow child node in the relation, then that is stored in this field.

   For e.g. bas:null:null:null: ਤੇਂ:null:null:N:null .ਸੱਤ ਦਸ **ਤੋਂ** ਛੋਟਾ ਹੈ

6. Positive Conditions for Parent: All the attributes and properties whose presence we want to assert about Parent node are stored in this field. obj:null:null:null: ਦੇ ਲਈ:ADJ:null:N:null. ਰਾਸ ਸੀਤਾ **ਦੇ ਲਈ** ਉਥੇ ਗਾਯਾ

7. Negative Conditions for Parent: All the attributes and properties whose absence we want to assert about Parent node are stored in this field.

8. Positive Conditions for Child: All the attributes and properties whose presence we want to assert about Child node are stored in this field. For e.g.

   cob:null:null:null: ਦੇ ਨਾਲ:null:null:N:null

9. Negative Conditions for Child: All the attributes and properties whose absence we want to assert about Child node are stored in this field.

For e.g. entry in database for bas relation is **"bas:null:null:null: ਤੇਂ:null:null:N:null"** which signifies that name of relation is bas and there is no case marker preceding or following the parent and also there is no case marker preceding the child. Case marker ਤੇਂ follows the child node. There are no that attributes, assert parent. And there are no attribute that should be absent for parent .the attributes that asserts child are N and there is no negative attribute for child. Now let us case marker for relations in detail, considering some of their properties.

agt: Agent i.e. a thing which initiates an action

Noun	ਨੇ	agt:null:null:null:ਨੇ:@past#V:VINT:N:null	ਰਾਮ **ਨੇ** ਚੌਲ ਖਾਦੇ
Pronoun	ਨੇ	agt:null:null:null:ਨੇ:@past#V:VINT:PRON:null	

and: conjunction i.e a conjunctive relation between concepts

And	ਤੇ	and:null:ਤੇ:null:null:null:null:null:null	ਰਾਮ **ਤੇ** ਸ਼ਾਮ ਦੋਸਤ ਹਨ

aoj : a thing which is in a state or has an attribute

Aoj	null	aoj:null:null:null:null:null:null:null:null	ਲਾਲ ਕਲਮ

bas basis i.e. indicates a thing used as the basis (standard) of comparison

Pronoun	ਤੋਂ	bas:null:null:null: ਤੋਂ:null:null:N:null	ਸੱਤ ਦਸ **ਤੋਂ** ਛੋਟਾ ਹੈ
Noun	ਤੋਂ	bas:null:null:null: ਤੋਂ:null:null:PRON:null	

ben beneficiary i.e indicates an indirectly related beneficiary or victim of an event or state

Noun	ਦੇ ਲਈ	ben:null:null:null: ਦੇ ਲਈ:null:null:N:null	ਦੇਸ਼ **ਦੇ ਲਈ** ਜਾਨ ਦੇਨਾ
pronoun	ਦੇ ਲਈ	ben:null:null:null: ਦੇ ਲਈ:null:null:PRON:null	

cag co-agent i.e. indicates a thing not in focus that initiates an implicit event that is done in parallel

Noun	ਦੇ ਨਾਲ	cag:null:null:null: ਦੇ ਨਾਲ:null:null:N:null	ਛੂਰੀ **ਦੇ ਨਾਲ** ਕੱਟੋ
Pronoun	ਦੇ ਨਾਲ	cag:null:null:null: ਦੇ ਨਾਲ:null:null:PRON:null	

cao co-thing with attribute i.e indicates a thing not in focus that is in a parallel state

Noun	ਦੇ ਨਾਲ	cao:null:null:null: ਦੇ ਨਾਲ:null:null:N:null	ਰਾਮ **ਦੇ ਨਾਲ** ਟੂਰੋ
Pronoun	ਦੇ ਨਾਲ	cao:null:null:null: ਦੇ ਨਾਲ:null:null:PRON:null	

cnt content i.e. indicates the content of a concept

Cnt	null	cnt:null:null:,:,:null:null:null:null	

cob affected co-thing

Noun	ਤੋਂ	cob:null:null:null: ਦੇ ਨਾਲ:null:null:N:null	ਰਾਮ ਸਾਮ **ਦੇ ਨਾਲ**
Pronoun	ਤੋਂ	cob:null:null:null: ਦੇ ਨਾਲ:null:null:PRON:null	ਵਾਰਦਾਤ ਿਵਚ ਜਖਮੀ ਹੋਇਆ

con condition i.e. indicates a non-focused event or state that conditions a focused event or state

con	ਜੇ	con: ਜੇnull: ਜੇ:null:null:null:null:null	

coo effected co-thing I.e. indicates a co-occurrent event or state for a focused event or state

coo	ਉਦੋ	coo: ਉਦੋਂ:null:ਜਦੋਂ:null:null:null:null:null	ਜਦੂੰ ਗਰਮ ਹੋਦਾਂ ਹੈ **ਉਦੂ** ਲਾਲ……

dur duration i.e indicates a period of time during which an event occurs or a state exists

dur	ਦੇ ਵੇਲੇ	dur:null:null:null:ਦੇ ਵੇਲੇ:null:null:null:null	ਸਵੇਰ **ਦੇ ਵੈਲੇ**:……

equ effected co-thing i.e. indicates an equivalent concept

equ	null	equ:null:null:,:,:null:null:null:null	

fmt range/from-to i.e. indicates a range between two things

Noun	ਤੋਂ	fmt:null: ਤਕ:null:ਤੋਂ:null:null:N:null	ਪਿਟਯਾਲਾ **ਤੋਂ** ਜਪਾਨ **ਤਕ**
Pronoun	ਤੋਂ	fmt:null: ਤਕ:null:ਤੋਂ:null:null:PRON:null	

frm origin i.e indicates an initial state of a thing or a thing initially associated with the focused thing

Noun	ਤੋਂ	frm:null:null:null:ਤੋਂ:null:null:N:null	ਪਿਟਯਾਲਾ **ਤੋਂ**…..
Pronoun	ਤੋਂ	frm:null:null:null:ਤੋਂ:null:null:PRON:null	

gol goal state i.e. indicates a final state of object or a thing finally associated with the object of an event

Noun	ਿਵਚ	gol:null:null:null: ਿਵਚ ਂ:null:null:INANI#N:null	
Pronoun	ਿਵਚ	gol:null:null:null:ਿਵਚੇਂ:null:null:INANI#PRON:null	
Noun	ਨੂ	gol:null:null:null: ਨੂ:null:null:N:null	
Pronoun	ਨੂ	gol:null:null:null: ਨੂ:null:null:PRON:null	

icl included/a kind of i.e. indicates an upper concept or a more general concept

icl	ਤਰਾਂ ਦਾ	icl:null:null: ਤਰਾਂ ਦਾ:null:null:null:null:null	ਕੁੱਤਾ ਇਕ **ਤਰਾਂ ਦਾ** ਜਾਨਵਰ ਹੈ

ins instrument i.e. indicates an instrument to carry out an event

Noun	ਤੋਂ	ins:null:null:null: ਤੋਂ/ਦੇ ਨਾਲ :null:null:N:null	ਕਲਮ **ਦੇ ਨਾਲ** ਿਲਖੋ
Pronoun	ਤੋਂ	ins:null:null:null: ਤੋਂ/ਦੇ ਨਾਲ :null:null:PRON:null	

int intersection i.e. indicates all common instances to have with a partner concept

int	null	int:null:null:null:null:null:null:null:null	-

iof an instance of i.e. indicates a class concept that an instance belongs to

iof	null	iof:null:null:null:null:null:null:null:null	ਪਿਟਯਾਲਾ ਪੰਜਾਬ ਦਾ ਸ਼ਿਹਰ ਹੈ

man manner i.e. indicates a way to carry out an event or the characteristics of a state

man	null	man:null:null:null:null:null:null:null:null	ਜਲਦੀ ਚਲੋ

met method or means i.e indicates a means to carry out an event

Noun	ਦੇ ਨਾਲ	met:null:null:null: ਦੇ ਨਾਲੋ :null:null:N:null	ਚਾਕੂ **ਦੇ ਨਾਲ** ਕੱਟੋ
Pronoun	ਦੇ ਨਾਲ	met:null:null:null: ਦੇ ਨਾਲ :null:null:PRON:null	

mod modification i.e. indicates a thing that restricts a focused thing

Noun	ਦਾ	mod:null:null:null:ਦਾ:@def:null:N:null	ਕਹਾਣੀ **ਦਾ** ਿਹੱਸ੍ਾ
Pronoun	ਦਾ	mod:null:null:null:ਦਾ:null:null:PRON:null	
Noun	ਦੇ ਨਾਲ	mod:null:null:null: ਦੇ ਨਾਲ:@topic:null:N:null	

Pronoun	ਦੇ ਨਾਲ	mod:null:null:null:ਦੇਨਾਲ:@topic:null:PRON:null	
Noun	ਦਾ	mod:null:null:null:ਦਾ:null:null:N:null	ਕਹਾਣੀ **ਦਾ** ਿਹੱਸ੍ਾ
Pronoun	ਦਾ	mod:null:null:null: ਦਾ:null:null:PRON:null	

nam name i.e. indicates a name of a thing

nam	ਤੇ	an nam:null:null:null:null:null:null:null:null	ਇਹ “ਰਾਮ” ਹੈ

obj affected thing i.e. indicates a thing in focus that is directly affected by an event or state

Noun	ਤੋਂ	obj:null:null:null: ਤੋਂ:V#link:RELagt:N:null	ਰਾਸ ਨੇ ਸੀਤਾ **ਤੋਂ** ਕੰਮ ਕਰਾਯਾ
Pronoun	ਤੋਂ	obj:null:null:null: ਤੋਂ:V#link:RELagt:PRON:null	
Noun	ਨੂੰ	obj:null:null:null: ਨੂੰ:V:VINT:ANIMT#N:null	ਰਾਸ ਨੇ ਸੀਤਾ **ਨੂੰ** ਫੁਲ ਦੀਤਾ
Pronoun	ਨੂੰ	obj:null:null:null:ਨੂੰ:V:VINT:ANIMT#PRON:null	
Noun	ਦੇ ਲਈ:	obj:null:null:null: ਦੇ ਲਈ:ADJ:null:N:null	ਰਾਸ ਸੀਤਾ **ਦੇ ਲਈ** ਉੂਥੇ ਗਾਯਾ
Pronoun	ਦੇ ਲਈ:	obj:null:null:null: ਦੇ ਲਈ:ADJ:null:PRON:null	

opl affected place i.e. indicates a place in focus affected by an event

Noun	ਿਵਚ/ਤੇ	opl:null:null:null: ਿਵਚ/ਤੇ :null:null:N:null	ਿਵਚਕਾਰ **ਤੋਂ** ਕਟ
Pronoun	ਿਵਚ/ਤੇ	opl:null:null:null: ਿਵਚ/ਤੇ :null:null:PRON:null	

or disjunction i.e. indicates a partner to have disjunctive relation to

or	ਜਾਂ	or:null:null:null: ਜਾਂ:null:null:null:null	ਰਾਮ **ਜਾਂ** ਸਾਮ….

per proportion/rate/distribution i.e. indicates a basis or unit of proportion, rate or distribution

per	ਵਾਰ	per:null:null:null:ਵਾਰ :null:null:null:null	ਇਕ ਿਦਨ ਿਵਚ ਅੱ ਵਾਰ…

plc place i.e. indicates a place where an event occurs, or a state that is true, or a thing that exists

Noun	ਿਵਚ	plc:null:null:null: ਿਵਚ◌ं:null:null:N:null	ਰਸੋਇ **ਿਵਚ** ਬਨਾ….
Pronoun	ਿਵਚ	plc:null:null:null: ਵਿਚ◌ं:null:null:PRON:null	

plf initial place i.e. indicates a place where an event begins or a state that becomes true

plf	ਤੇ	plf:null:null:null: ਤੋਂ :null:null:null:null	ਪਿਟਯਾਲਾ **ਤੋਂ** ਜਪਾਨ ਤਕ

plt final place i.e. indicates a place where an event ends or a state that becomes false

plt	ਤਕ	plt:null:null:null: ਤਕ :null:null:null:null	ਪਿਟਯਾਲਾ **ਤਕ** ਸਫਰ..

pof part of i.e. indicate a concept of which a focused thing is a part

Noun	ਦਾ/ਦੀ/ਦ◌े	pof:null:null:null: ਦਾ/ਦੀ/ਦ◌े :null:null:N:null	ਿਕਤਾਬ **ਦੀ** ਰਵਾਤ…
Pronoun	ਦਾ/ਦੀ/ਦ◌े	pof:null:null:null: ਦਾ/ਦੀ/ਦ◌े :null:null:PRON:null	

pos possessor i.e. indicates the possessor of a thing

pos	null	pos:null:null:null:null:null:null:null:null	ਮੇਰੀ ਿਕਤਾਬ

ptn partner i.e. indicates an indispensable non-focused initiator of an action

Noun	ਦੇ ਨਾਲ:	ptn:null:null:null: ਦੇ ਨਾਲ:null:null:N:null	ਰਾਮ **ਦੇ ਨਾਲ** ਸਾਮ..

Pronoun	ਦੇ ਨਾਲ:	ptn:null:null:null: ਦੇ ਨਾਲ:null:null:PRON:null	

pur purpose i.e. indicates the purpose or objective of an agent of an event or the purpose of a thing that exists

Noun	ਦੇ ਲਈ	pur:null:null:null: ਦੇ ਲਈ :null:null:N:null	ਰਾਮ ਨੂੰ ਬਚਾਉਨ **ਦੇ ਲਈ**….
Pronoun	ਦੇ ਲਈ	pur:null:null:null: ਦੇ ਲਈ :null:null:PRON:null	

qua quantity i.e. indicates the quantity of a thing or unit

qua	null	qua:null:null:null:null:null:null:null:null	**ਦੋ** ਚੂਹੇ

rsn reason i.e. indicates a reason why an event or a state happens

rsn	ਦੇ ਕਰਨ	rsn:null:null:null ਦੇ ਕਰਨ:null:null:null:null	ਮੀਂਹ **ਦੀ ਕਰਨ** …

scn scene i.e. indicates a scene where an event occurs, or state is true, or a thing exists

Noun	ਿਵਚ	scn:null:null:null: ਿਵਚ ਂ:null:null:N:null	…..ਟੀ ਵੀ **ਿਵਚ** ਆਇਆ
Pronoun	ਿਵਚ	scn:null:null:null: ਿਵਚ :null:null:PRON:null	

seq sequence i.e**.** Indicates a prior event or state of a focused event or state

Noun	ਦੇ ਬਾਅਦ	seq:null:null:null: ਦੇ ਬਾਦ :null:null:N:null	ਤੀਨ **ਦੇ ਬਾਅਦ** ਚਾਰ
Pronoun	ਦੇ ਬਾਅਦ	seq:null:null:null: ਦੇ ਬਾਦ :null:null:PRON:null	

src indicates the initial state of an object or thing initially associated with the object of an event

Noun	ਤੋਂ	src:null:null:null: ਤੋਂ :null:null:N:null	ਬਤੀ ਲਾਲ **ਤੋਂ** ਰੀ……
Pronoun	ਤੋਂ	src:null:null:null: ਤੋਂ:null:null:PRON:null	

tim time i.e. indicates the time an event occurs or a state is true

Noun	ਨੂੰ	tim:null:null:null: ਨੂੰ:null:null:N:null	ਸੰਗਲਵਾਰ **ਨੂੰ**
Pronoun	ਨੂੰ	tim:null:null:null: ਨੂੰ:null:null:PRON:null	

tmf initial time i.e indicates the time an event starts or a state becomes true

Noun	ਤੋਂ	tmf:null:null:null:ਤੋਂ :null:null:N:null	….ਸਵੇਰ **ਤੋਂ** ਲੇਕੇ ਸਾਮ
Pronoun	ਤੋਂ	tmf:null:null:null:ਤੋਂ :null:null:PRON:null	ਤਕ

tmt final time i.e indicates a time an event ends or a state becomes false

Noun	ਤਕ	tmt:null:null:null: ਤਕ:null:null:N:null	….ਸਵੇਰ ਤੋਂ ਲੇਕੇ ਸਾਮ
Pronoun	ਤਕ	tmt:null:null:null: ਤਕ:null:null:PRON:null	**ਤਕ**

to

Noun	ਤੋਂ	to:null:null:null: ਦੇ ਨਾਲ:null:null:N:null	
Pronoun	ਤੋਂ	to:null:null:null: ਦੇ ਨਾਲ:null:null:PRON:null	

via an intermediate place or state

Noun	ਵਲੋਂ	via:null:null:null:ਤੋਂ ਹੋਕੇ :null:null:N:null	ਪਿਟਯਾਲਾ **ਵਲੋਂ**…..
Pronoun	ਵਲੋਂ	via:null:null:null :ਵਲੋਂ:null:null:PRON:null	

This database is stored in a text file and is referred by system while conversion to target language i.e. Punjabi.

4.6 Morphology

Morphology is the study of the minimal grammatical units of a language and of their formation into words, including inflection, derivation and composition[13]. Morpheme that is the main concern of morphology is "a minimal unit of meaning or grammatical. function". Morphemes can be divided into two parts:

- **Free Morphemes** are those that can occur alone or which can stand on their own are called free morphemes. E.g. ਗੁਲਾਬ, ਿਬੱਲੀ
- **Bond Morphemes** are those, which do not occur alone. All affixes are bond morphemes. E.g. ਅਨ -ਅਨਪੜ, ਹਾਰ -ਹੋਨਹਾਰ

The general meaning of the morphology is study of the word. To get the more natural meaning of the word, the word should be changed or something should be removed or added to get the complete sense of the sentence. In addition to case marker there are some words that get changed according to the sentence [5].

E.g. ਬੱਚਾ ਖਾੱਣਾ ਖਾ ਰਹਾ ਹੈ।

ਬੱਚੀ ਖਾੱਣਾ ਖਾ ਰਹੀ ਹੈ।

The difference between the two sentences can be easily seen. In the first sentence a boy is eating food while in second a girl is eating food. ਬੱਚ becomes ਬੱਚਾ in case of male while becomes ਬੱਚੀ in case of female. Similarly with ਰਹ while becomes ਰਹਾ and ਰਹੀ respectively. This is called morphology of the word i.e. changing the word according to sentence [6].

The morphology can be categorized into three types:

- Attribute Label Resolution Morphology
- Relation Label Resolution Morphology
- Noun, Verb and Adjective Morphology

We are mainly dealing in this project with Attribute Label Resolution Morphology and Relation Label Resolution Morphology. Noun, Verb and Adjective Morphology is not being explicitly dealt. And there will be correct morpheme produced many a times for

nouns, verbs and adjectives just by matching the attributes (like tense, number, aspect etc.) associated with the UW. But we have given its separate description for the purpose of overall understanding.

4.6.1 Attribute Label Resolution Morphology

Attribute label resolution deals with determining the Punjabi equivalent of attribute labels in a node. Attribute label resolution may introduce new words as is the case while referring to definitive and in definitive articles. Attribute label resolution may also change the form of the word depending on the tense, number or gender of the word or node.

Suffix	*Tense*	***Aspect***	***Mood***	***Number***	***Gender***	***Person***	***Vowel Ending***
ਦਾ ਹਾਂ	@present	@custom	-	sg	male	1st	-
ਦਾ ਸੀ	@past	@custom	-	sg	male	-	-
◌ਾ	@past	@complete	-	sg	male	-	a
◌ੀ	@past	@complete	-	@pl	female	-	a
◌ਾਗੀਂ	@future	-	-	sg	female	1st	a
◌ਾਗੋਂ	@future	-	-	@pl	male	3rd	e
◌ਾਗੀਆਂ	@future	-	-	@pl	female	-	A

table 4.1 attribute label resolution morpholgy

4.6. 2 Relation Label Resolution Morphology

The resolution of a relation label depends on the presence or absence of certain properties of the node that undergoes a change and its parent(s) nodenet. The rule-base is basically a set of conditions to be satisfied by a node and its parent along with the corresponding action that is to be taken in the form of suffixing or prefixing a morpheme. In some cases, a new word may be inserted (such as jisda) representing clausal information[14].

For Example:

English sentence: Ram who loves Sita saved Shyam.

Punjabi equivalent: us ram ne Shyam nu bachya **jehra** sita nu pyar karda hai.

The words **jehra and us** are added here [6]. This kind of morphology is dealt with under the section of "casemarkers" except one special case of "aoj" relation label, which is explained in a later section.

4.6.3 Noun, Verb and Adjective Morphology

Relation label and Attribute label resolution have brought the form of the sentence very close to its final form, but it's not quite there yet. Some morphology, which depends on the phonetic properties of the Punjabi word in question, has to be considered.

4.6.3.1 Noun Morphology

Noun morphology is the study of nouns in the sentence. Depending upon the gender, number and vowel ending of the noun, some part of the word is removed from the end and a new phoneme is added to the end of the word.

Some example sentences of noun morphology usage:

1. When the noun has the gender male and ends with vowel 'A' and also if it has suffix and it is in plural form then at the end of the root word ੁਂ is added. E.g. ਬੱਚੇਂ ਨੇ ਪੜਾਈ ਕੀ।

Here ਬੱਚ is the root word, originally it is ਬੱਚਾ, therefore the word will have the lexical attribute NA (Noun ending with A) and also it is male noun and represent in the plural form with suffix ਨੇ, hence ੇ should be added at the end of the root word.

2. When the noun has the gender male and ends with vowel 'A' and also if it has suffix but it is in singular form then at the end of root word ' ੇ ' should be added. E.g.

ਬੱਚੇ ਨੇ ਪੜਾਈ ਕੀ।

4.6.3.2 Adjective Morphology

The adjectives in the dictionary are given for male gender. Some adjectives change their form according to their use. To distinguish them from the non-changeable adjectives, they are given the lexical attribute AjdA. Only these adjectives change according to the context and hence the last vowel A is deleted from the dictionary entry. E.g.

ਚੰਗਾ ਲੜਕਾ, ਚੰਗੀਂ ਲੜਕੀ, ਚੰਗੇਂ ਲੜਕੇ

In the above example, the adjective ਚੰਗ changes according to its use. It ends with the vowel ਆ therefore it has the lexical attribute AdjA. Since ਾ changes into ੀ, ੇ it will not be present in the dictionary entry of that adjective. Change in adjective depends on the gender, plural/singular form of gender, suffix with gender etc.

4.6.3.4 Verb Morphology

Verbs are essential to the clause structure and can inflect and show contrasts of tense, aspect, mood, voice, number and person. Verbs are usually subdivided into two parts: Lexical/Main verbs and Auxiliary verbs. Main verbs are further classified as Transitive verbs and Intransitive verbs [6].

Main verb morphology

Verb morphology is the study of verb forms in the sentence. It plays a very important role in generating a correct sentence. Consider the following example

Simple Present

ਲੜਕੇ ਖੇਡਦੇ ਨੇ।

ਲੜਕੇ ਖਾਣਦੇ ਨੇ।

Simple Past

ਲੜਕੋਂ ਨੇ ਖੇਿਡਆ

ਲੜਕੋਂ ਨੇ ਖਾੇਿਯਆ

Here in both sentences two different verbs are used. Though verb in first sentence is ਖੇੇਿਡ ending with vowel ਅ and verb in second sentence is ਖਾ ending with vowel ਆ, verb morphology is same for the both verb i.e. ਦੇ ਨੇ. Whereas, in the case of simple past tense, the verb ਖੇਲ ending with last vowel ਅ has become ਖੇੇਿਡਆ and the verb ਖਾ has become ਖਾਿਯਆ.

Auxiliary Verb Morphology

When main verb is not present in the sentence then auxiliary verb acts like the main verb. Each UNL representation of a sentence will have a main verb, but in case of attributive adjective, verb terminator is required. These are actually auxiliary verbs. When UW1 of aoj relation has the attribute @pred then it is considered as predicative adjective
and it requires verb terminator. Let see the following examples

ਤੁਸੀ ਬੜੇ ਚੰਗੋਂ ਹੋ।

ਤੁਸੀ ਚੰਗੋਂ ਰਹੋਗੇ।

We can see that according to the tense, gender, number etc. there is a verb terminator required to complete the sentence. Thus above examples give the fair idea of Auxiliary verb morphology. The section below on implementation of the morphology system again pours light on some peculiarities of this case.

4.6.4 Implementation of Morphology Rule Extraction

Morphology Rule file is implemented using the algorithm given below which uses various files such as link, lnk,ja etc.One of the main attribute used here is AttributeCloseness.AttributeCloseness parameter is nothing but the number of attributes matched between UW and morpheme.

Algorithm Flow

```
for (each UW to be examined)
repeat
  AttributeList ← all attributes associated with a UW
  if(UW has a attribute "@link")
     start scanning morphology rulebaseforlink;
     calculate AttributeCloseness param. with each entry;
     if (AttributeCloseness > 0)
        return morpheme with the maximum value;
    else
        return default value;
  else if(UW has a attribute "@lnk")
     start scanning morphology rulebaseforlnk;
     calculate AttributeCloseness param. with each entry;
     if (AttributeCloseness > 0)
        return morpheme with the maximum value;
    else
        return default value;
  else if(UW has a attribute "@jA")
     start scanning morphology rulebaseforjA;
     calculate AttributeCloseness param. with each entry;
     if (AttributeCloseness > 0)
        return morpheme with the maximum value;
    else
        return default value;
  else if ((UW has a attribute "@entry") &&
           (UW is connected to its child with "aoj" relation))
     start scanning morphology rulebase1;
     calculate AttributeCloseness param. with each entry;
     if (AttributeCloseness > 0)
        return morpheme with the maximum value;
     else
        return default value;
  else
     start scanning morphology rulebase2;
     calculate AttributeCloseness param. with each entry;
```

```
if (AttributeCloseness > 0)
   return morpheme with the maximum value;
else
   return default value;
```

Description of Algorithm

In this algorithm we take the attributes of the given UW in the parameter Attributelist and compare it with the attributes of the various files such as link,lnk ,etc and then calculate the value of th parametre attribute clossness ,if its greater then 0 then corresponding morpheme is returned else default value is returned.

4.7 Implementation Details

The important fuction used by generator system are the following

1) node.hindi() : This function involves a reference to Hindi Master Dictionary where by the hindi string and all other lexical information is extracted for the Universal Word.

2) caseMark(relationName, parentNode, childNode) This file involves reference to the Case Marker database and case markers for the parent and child are looked upon and added if required.

3) processWordRelation(): this function is called if the UNL document contains a universal word ([W] [/W]). This function updates scopeList, relationList and nodeList

4) processSentenceRelation(): this function is used to process a line if the line is identified as a binary relation [rel(uw1,uw2)]. This process also updates nodeList, scopeList, relationList.

5) getNearestOrphan(): In cases where the entry node is not an orphan and has multiple parents, the traversal is started from the nearest ancestor of the entry node which is an orphan. This function realizes our objective by performing a BFS starting from the entry node and following parent lists for expansion (instead of children list.

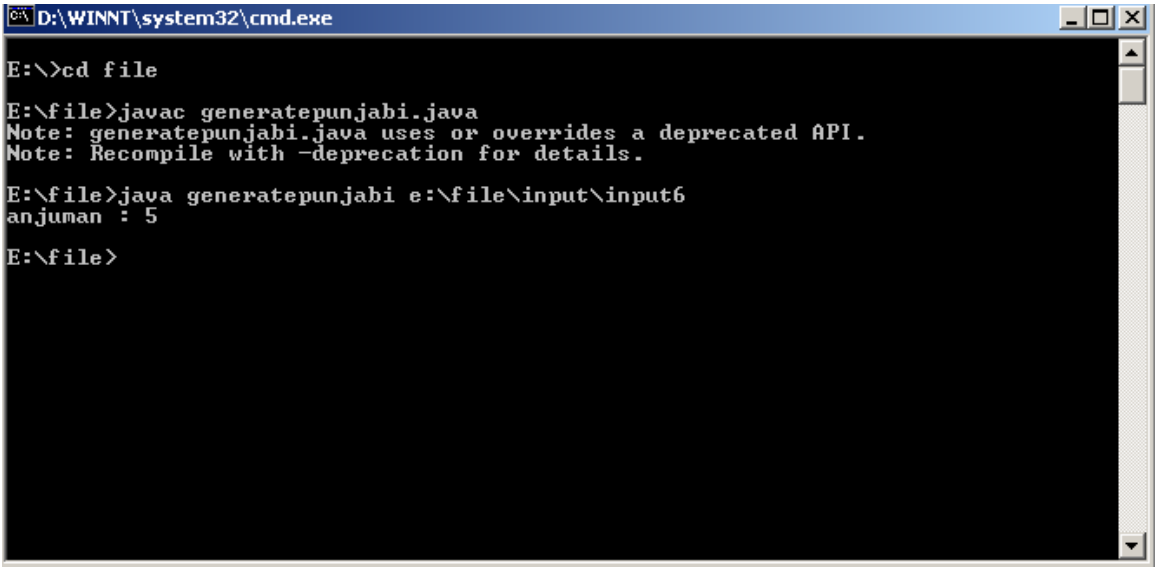
6) syntaxPlanNoScope() : This is function that realizes basic Syntax Plan of simple Sentence with no scopes and multiple-parents. It contains all the heuristics and a word order based on these heuristics is generated. Given a node this recursive function returns the complete word order of all uws that are reachable from this node, it remains oblivious of other orphans in the same scope and when it encounters a scope node it offers no special treatment (this is later dealt by syntaxPlanScope)

7) syntaxPlanScope() : This function is essentially the cover function written to care of complex and compound sentences. This is a recursive function which is used for getting syntax plan in a scope and it returns the complete word order of all uws of that scope, it calls syntaxPlanNoScope() on all the orphans of this scope then combines them and then it checks for compound-uw and recursively expands those scope nodes in their place.

Chapter 5: Experimentation

This chapter presents the result of UNL to Punjabi deconversion using the system. For experimentation we have use simple sentences, clausal sentences and sentences from the ITU corpus.

We executing the software for the given input file and get the corresponding output in the file finalout.txt. Now let us conside few sentences.



5.1 Generation for the individual sentence

1. **Input**: - Bird fly

   UNL:
   {unl}
   agt(fly(icl>do).@entry.@pres.@generic, bird(icl>fauna))
   {/unl}

   Output: ਉਡਿਅਾ ਪੰਛੀ

   Desired Output: ਪੰਛੀ ਉਡਿਅਾ

2. **Input**: Rita dances well

UNL:

{unl}
agt(dance.@entry,Rita)
man(dance,well)
{/unl}

Output: ਰੀਟਾ ਚੰਗਾ੍ ਨੱਚਨਾ੍

Desired Output: ਰੀਟਾ ਚੰਗਾ ਨੱਚਦੀ ਰੈ

3. **Input**: John wants pen

UNL:

{unl}
agt(want(agt>thing,obj>thing).@entry.@present, John)
obj(want.@entry.@present, pen(icl>writing instrument))
{/unl}

Output: ਜੋਨ ਕਲਮ ਚਾਹ ਹੈ

Desired Output: ਜੋਨ ਨੂੰ ਕਲਮ ਚਾਹੀਦੀਂ ਹੈ

4. **Input:** He is talented by birth

UNL:

{unl}
agt(talent(icl>person).@entry.@past, he)
tmf(talent,birth(icl>phenomenon))
{/unl }

Output: ਉਹ ਜਨਸ ਤੋਂ ਪ੍ਰਿਤਭਾਸ਼ਾਲੀ

Desired Output: ਉਹ ਜਨਸ ਤੋਂ ਪ੍ਰਿਤਭਾਸ਼ਾਲੀ ਹੈ

5. **Input**: Ram was teacher between 1980 and 1984

UNL:

{unl}
agt(teacher.@entry, Ram)
tmf(teacher,1980)
tmf(teacher,1984)
{/unl}

Output: ਰਾਮ 1980 1984 ਅਿਧਆਪਕ ਸੀ

Desired Output: ਰਾਮ 1980 ਤੋਂ 1984 ਤਕ ਅਿਧਆਪਕ ਸੀ

6. **Input:** Ram eat rice with spoon in Delhi

UNL:
{unl}
agt(eat(icl>do).@entry, Ram)
obj(eat,rice)ins(eat,spoon)plc(eat,Delhi)
{/unl}

Output: ਰਾਮ ਿਦੱਲੀ ਚਮਚ ਤੋਂ/ਦੇ ਨਾਲ ਚੋਲ ਖਾਣਾ

Desired Output: ਰਾਮ ਿਦੱਲੀ ਿਵਚ ਚਮਚ ਦੇ ਨਾਲ ਚੋਲ ਖਾਂਦਾ ਹੈ

7. **Input**:I saw a leopard in the campus

UNL:
{unl}
agt(see(icl>perceive(agt>volitional thing,obj>thing)).@entry.@past, I)
obj(see.@entry.@past,leopard(icl>panther).@pl)
plc(see(icl>perceive(agt>volitional thing,obj>thing)), campus(icl>institution))
{/unl}

Output: ਮੈਨੇ ਇਹਾਤਾ ਿਵਚ ਚੀਤਾ ਵੇਖਾ

Desired Output: ਮੈਨੇ ਇਹਾਤਾ ਿਵਚ ਚੀਤਾ ਵੇਖਾ

8. **Input:**He was beaten badily

UNL:
{unl}
obj(beat.@entry.@past,he)man(beat, badly)
{/unl}

Output: ਉਹ ਨੂੰ ਮਾਰ ਰਹਾ ਸੀ JAB/JO/JAHA ਬੁਰੀ ਤਰਹ

Desired Output: ਉਹ ਨੂੰ ਬੁਰੀ ਤਰ੍ਹਾਂ ਮਾਿਰਆ ਿਗਆ

9. **Input:** Oe has travelled abroad widely.

UNL:
{unl}
man(travel(agt>thing):07.@entry.@complete, widely:0O)
agt(travel(agt>thing):07.@entry.@complete, Oe(iof>person):00.@topic)
man(travel(agt>thing):07.@entry.@complete, abroad:0H)
{/unl}

Output: Oe ਿਵਦੇਸ਼ ੍ ਦੂਰ ਦੂਰ ਤਕ ਯਾਤਰਾ ਿਕਤਾ ਹੈ

Desired Output: Oe ਨੇ ਿਵਦੇਸ਼ਾਂ੍ ਿਵਚ ਦੂਰ ਦੂਰ ਤਕ ਯਾਂਤਰਾ ਿਕਤੀ

10. Input: Poet abc wrote beautiful poem

UNL:
{unl}
agt(write(agt>thing,obj>thing).@entry,poet(icl>person))
obj(write,poem)nam(poet,abc)aoj(poem,beautiful)
{/unl}

Output: ਕਵੀ੍ abc ਸੁੰਦਰ ਕਿਵਤਾ ਿਲਖਾ

Desired Output: ਕਵੀ੍ abc ਸੁੰਦਰ ਕਿਵਤਾ ਿਲੱਖੀ

11. Input:Ram eat rice in afternoon at his home with spoon

UNL:
{unl}
agt(eat(icl>do).@entry,Ram)
obj(eat,rice)
tim(eat(icl>do).@entry,afternoon)
plc(eat(icl>do),home)
ins(eat,spoon)
{/unl}

Output: ਰਾਮ ਲੋਢੇ ਵੇਲੇ ਘਰ੍ ਿਵਚ◌ਂ ਚਮਚ ਤੋਂ◌ੇ/ਦੇ◌ੇ ਨਾਲ ਚੌਲ ਖਾਣਾ

Desired Output: ਰਾਮ ਘਰ੍ ਿਵਚ◌ਂ ਲੋਢੇ ਵੇਲੇ ਚਮਚ ਨਾਲ ਚੌਲ ਖਾਂਦਾ ਹੈ

12 Input:Japanese novelist won Nobel Prize in 1954

UNL:
{unl}
aoj(Japanese(aoj>thing).@entry,novelist(icl>person))
agt(win(icl>get(agt>thing,obj>thing)),novelist)
agt(win(icl>get(agt>thing,obj>thing)),Nobel Prize)
tim(win,1954)
{/unl}

Output: ਨਾਵਲਕਾਰ੍ JAB/JO/JAHA ਨੋਬਲ ਪੁਰਸਕਾਰ 1954 ਪਾ ਯਾ ਜਪਾਨੀ ਹੈ

Desired Output: ਜਪਾਨੀ ਨਾਵਲਕਾਰ੍ ਨੇ 1954 ਿਵਚ ਨੋਬਲ ਪੁਰਸਕਾਰ ਿਜਤਆ

13. Input: Ram happily eats rice with spoon in evening in a hotel.

UNL:
{unl}
agt(eat(icl>do).@entry,ram)
obj(eat.@entry,rice)
ins(eat.@entry,spoon)

man(eat.@entry,happily)
tim(eat.@entry,evening)
plc(eat.@entry,hotel)
{/unl}

Output: ਰਾਮ ਖੂਸ਼ੀ ਸ਼ਾਮ ਤੋਂ ਹੋਟੱਲ ਿਵਚ ਚਮਚ ਦੇ ਨਾਲ ਚੋਲ ਖਾਣਾ

Desired Output: ਰਾਮ ਸ਼ਾਮ ਨੂੰ ਹੋਟੱਲ ਿਵਚ ਖੁਸ਼ੀ ਚਮਚ ਨਾਲ ਚੌਲ ਖਾਂਦਾ ਹੈ

14. **Input:** Japanese novelist who was awarded the Nobel Prize for literature in 1994.

UNL:
{unl}
aoj(Japanese(aoj>thing):00,novelist(icl>person):09.@entry)
gol(award(agt>thing,gol>thing,obj>thing):0Q.@past,novelist(icl>person):09.@entry)
obj(award(agt>thing,gol>thing,obj>thing):0Q.@past,Nobel Prize:12.@def)
tim(award(agt>thing,gol>thing,obj>thing):0Q.@past,1994:1W)
pur(Nobel Prize:12.@def,literature(icl>art):1I)
{/unl}

Output : ਜਪਾਨੀ ਨਾਵਲਕਾਰ ਨੂੰ JAB/JO/JAHA ਸਾਿਹਤ ਦੇ ਲਈ ਨੋਬਲ ਪੁਰਸਕਾਰ ਤੋਂ ੧੯੯੪ ਸੇਂ ਸਨਮਾਨਤ ਿਕਤਾ

Desired Output: ਜਪਾਨੀ ਨਾਵਲਕਾਰ ਨੇ ੧੯੯੪ ਿਵਚ ਸਾਿਹਤ ਲਈ ਨੋਬਲ ਪੁਰਸਕਾਰ ਤੋਂ ਸਨਮਾਨਤ ਿਕਤਾ ਿਗਆ

15. **Input:** Oe has often dealt with <c><c>marginal people and outcasts</c> and isolation from individual level to <c>social and cultural</c> levels</c>.

UNL:
{unl}
obj(deal with(agt>thing,obj>thing):0D.@entry.@complete.@past, :03)
agt(deal with(agt>thing,obj>thing):0D.@entry.@complete.@past,
Oe(iof>person):00.@topic)
man(deal with(agt>thing,obj>thing):0D.@entry.@complete.@past, often:07)
and:03(isolation(icl>event):1V.@entry, :02)
mod:03(isolation(icl>event):1V.@entry, :04)
and:02(outcast(icl>person):1E.@entry.@pl, people(icl>person):13)
aoj:02(marginal(aoj>thing):0R, people(icl>person):13)
fmt:04(level(icl>degree):3P.@entry.@pl,level(icl>degree):2P)
aoj:04(:01, level(icl>degree):3P.@entry.@pl)
and:01(cultural(aoj>thing):3C.@entry, social(aoj>thing):31)
mod:04(level(icl>degree):2P,individual(mod<thing):2E)
{/unl}

Output: Oe ਨੇ ਅਕਸਰ ਿਨੱਜੀ ਦਰਜੇ ਦੇ ਸੰਸਕ੍ਰਿਤ ਤੇ ਸਮਾਜਕ ਦਰਜਾ ਤਕ ਅੱਡਰੇਪਣ ਤੇ ਪਿਤਤ ਤੇ ਸੀਮਾਂਤ ਲੋਕ ਤੂੰ ਪੇਸ ਿਕਤਾ ਸੀ

Desired Output: Oe ਅਕਸਰ, ਸੰਸਕ੍ਰਿਤ ਤੇ ਸਮਾਿਜਕ ਦਰਜੇ ਤੋਂ ਅੱਡਰੇਪਣ , ਪਿਤਤ ਤੇ ਸੀਮਾਂਤ ਲੋਕਾਂ ਨੂੰ ਪੇਸ ਿਕਤਾ ਰੈ

5.2 Generation for the Clausal sentences

1. **Input**:John and mary are friends

UNL:

```
{unl}
aoj (friend(icl>releation).@pl.@entry.@present, :01)
and :01(Mary, John)
{/unl}
```

Output: ਜੇਨ ਤੇ ਮੇਰੀ ਅਾੜੀ

Desired Output: ਜੇਨ ਅਤੇ ਮੇਰੀ ਅਾੜੀ ਹਨ

2. **Input:**From nobel Lecture, 1994

UNL:

```
{unl}
[W]
:01.@entry
[/W]
obj:01(from(icl>how(obj>thing)):01.@entry,    Nobel Lecture:06)
tim:01(Nobel Lecture:06,      1994:0L)
{/unl}
```

Output: ੧੯੯੪ ਿਵਚ ਨੋਬਲ ਿਵਆਿਖਆਨ ਤੋਂ

Desired Output: ੧੯੯੪ ,ਨੋਬਲ ਿਵਆਿਖਆਨ ਤੋਂ

3. **Input**:Ram who lives in Delhi eat rice

UNL:

```
{unl}
agt(eat(agt>person,obj>food).@entry,:01)
obj(eat(agt>person,obj>food), rice(icl>food))
agt:01(live(agt>person), Ram)
 plc:01(live(agt>person), Delhi(icl>person))
{/unl}
```

Output: ਰਾਮ ਿਦੱਲ਼ੀ ਿਵਚ ਰਿਹੰਦਾਂ ਰੈ, ਚੋਲ ਖਾਣਾ

Desired Output: ਰਾਮ ਿਜਹੜਾ ਿਦੱਲ਼ੀ ਿਵਚ ਰਿਹੰਦਾਂ ਰੈ, ਚੋਲ ਖਾਂਦਾ ਹੈ

4. **Input**:Ram saw sita who lives in Delhi

UNL:

{unl}
obj(see(icl>perceive(agt>volitionalthing,obj>thing)).@entry.@past,:01)
agt(see(icl>perceive(agt>volitional thing,obj>thing)) , Ram)
agt:01(live(agt>person), Sita)
plc:01(live(agt>person), Delhi(icl>person))
{/unl}

Output: ਰਾਮ ਿਦੱਲ਼ੀ ਰਿਹੰਦਾਂ ਸੀਤਾ ਨੂ ਵੇਖਾ

Desired Output: ਰਾਮ ਨੇ ਸੀਤਾ ਨੂੰ ਵੇਿਖਆ ਜੋ ਜੇੜੀ ਿਦੱਲ਼ੀ ਿਵਚ ਰਿਹੰਦੀਂ ਰੈ

5. **Input:**Ram like singing and dancing UNL:

UNL

{unl}
agt(like(icl>action).@entry ,Ram)
obj(like,:03)and:03(sing(icl>do),dance)
{/unl}

Output: ਰਾਮ ਗਾਣਾ और ਨਚਨਾ ਪਸੰਦ

Desired Output: ਰਾਮ ਨੂੰ ਗਾਣਾ ਤੇ ਨਚਨਾ ਪਸੰਦ ਰੈ

6. **Input:** if you are tired we will go straight home

UNL:

{unl}
agt(go(agt>thing,gol>thing).@entry,we)
plc(go.@entry,home)
con(go,:01)
aoj:01(tried,you(aoj>person))
{/unl}

Output: ਅਸੀਂ ਘਰ ਿਵਚ ਜੇ ਤੁਸੀ ਥਕ ਜਾਂਦਾ ਹੈ

Desired Output: ਜੇ ਤੁਸੀ ਥਕ ਗਏ ਹੋ ਤਾਂ ਅਸੀਂ ਘਰ ਚਲਨੇ ਹੈ

7. **Input:**if I shout you will be irritated

UNL:

{unl}
obj(irritate.@entry.@past, you)
con(irritate ,:01)
agt:01(shout,I)
{/unl}

Output:ਤੁਸੀ ਮੈਂ ਿਚਲਾ ਤਾਂ ਿਚੜ

Desired Output: ਜੇ ਮੈਂ ਿਚਲਾੱਅਾ ਤਾਂ ਤੁਸੀ ਿਚੜ ਜਾਉਗੇਂ

8. **Input:** another central theme - as in the works of a number of other Japanese writers is the conflict between <c>traditions and modern western culture</c>.

UNL:

{unl}
aoj(conflict(icl>phenomenon):2F.@def.@entry, theme(icl>subject):0G.@topic)
mod(theme(icl>subject):0G.@topic, another(mod<thing):00)
mod(theme(icl>subject):0G.@topic, central(mod<thing):08)
aoj(as in(aoj>thing,obj>thing):0O, theme(icl>subject):0G.@topic)
obj(as in(aoj>thing,obj>thing):0O, work(icl>book):0Y.@def.@pl)
mod(work(icl>book):0Y.@def.@pl, writer(icl>person):1Y.@indef.@pl)
mod(writer(icl>person):1Y.@indef.@pl, number of(qua<thing):19)
mod(writer(icl>person):1Y.@indef.@pl, other(mod<thing):1J)
aoj(Japanese(aoj>thing):1P, writer(icl>person):1Y.@indef.@pl)
aoj(between(aoj>thing,obj>thing):2O,conflict(icl>phenomenon):2F.@def.@entry)
obj(between(aoj>thing,obj>thing):2O, :01)
and:01(culture(icl>abstract thing):3T.@entry, tradition(icl>custom):2Z.@pl)
mod:01(culture(icl>abstract thing):3T.@entry, modern(mod<thing):3E)
aoj:01(western(aoj>thing):3L, culture(icl>abstract thing):3T.@entry)
{/unl}

Output: ਇਕ ਰੋਰ ਕੇਂਦਰੀ ਮੂਲ ਿਵਸ਼ਾ ਕਈ ਹੋਰ ਜਪਾਨੀ ਲੇਖਕ ਦੀ ਰਚਨਾਵਾਂ ਿਜਵੇਂ ਿਕ ਆਧੁਿਨਕ ਪੱਛਮੀ ਸੰਸਕ੍ਰਿਤ और ਪਰਮਪਰੀਕ ਦੇ ਿਵਚ ਂ ਸੰਘਰਸ਼ ੍ ਹੈ

Desired Output: ਇਕ ਰੋਰ ਕੇਂਦਰੀ ਮੂਲ ਿਵਸ਼ਾ , ਿਜਵੇਂ ਿਕ ਕਈ ਹੋਰ ਜਪਾਨੀ ਲੇਖਕ ਦੀਆ ਰਚਨਾਵਾਂ ਹੈ , ਆਧੁਿਨਕ ਪੱਛਮੀ ਤੇ ਪਰਮਪਰੀਕ ਸੰਸਕ੍ਰਿਤ ਦੇ ਿਵਚ ਂ ਸੰਘਰਸ਼ ੍ ਹੈ

9. **Input**: Oe has once said.t "yeats is the writer in whose wake I would like to follow."

UNL:

{unl}
agt(say(agt>thing,obj>thing):0C.@complete.@entry, Oe(iof>person):00.@topic)
man(say(agt>thing,obj>thing):0C.@complete.@entry, once(icl>how):07)
obj(say(agt>thing,obj>thing):0C.@complete.@entry, :01)
aoj:01(writer(icl>person):0X.@def.@entry, Yeats(iof>poet):0K.@topic)
obj:01(in the wake of:1D, writer(icl>person):0X.@def.@entry)

agt:01(follow(agt>thing,obj>thing):1Y.@wish, I:1I)
man:01(follow(agt>thing,obj>thing):1Y.@wish, in the wake of:1D)
{/unl}

Output: Oe ਇਕ ਵਾਰੀ "ਯੇਟਸ ਲੇਖਕ ਹੈ ਮੈਂ ਦੇ ਪੀਛੇ ੇ ਅਨੁਸਰਣ ਹੈ " ਕਹੇ◌ਾ ਹੈ

Desired Output: Oe ਨੇ ਇਕਵਾਰ ◌ਿਕਹਾ ਸੀ "ਯੇਟਸ ਇਕ ਲੇਖਕ ਹੈ ◌ਿਜਸਂ ਦੇ ਪੀਛੇ◌ੈ ਮੈਂ ਅਨੁਸਰਣ ਕਰਨਾ ਚਾਹੁਦਾ ਹਾਂ "

10. Input: he won an admission to the university of Tokyo, where he <c>studied. french literature and received his B.A. in 1959</c>

UNL:
{unl}
obj(win(icl>get(agt>thing,obj>thing)):03.@entry.@past, admission(icl>action):0A.@indef)
agt(win(icl>get(agt>thing,obj>thing)):03.@entry.@past, he:00.@topic)
gol(admission(icl>action):0A.@indef, University of Tokyo:0R.@def)
plc(:01, University of Tokyo:0R.@def)
agt(:01, he:1I)
and:01(receive(agt>thing,obj>thing):2K.@entry.@past,
study(icl>learn(agt>thing,obj>thing)):1O.@past)
obj:01(receive(agt>thing,obj>thing):2K.@entry.@past, BA(equ>Bachelor of Arts):2X)
tim:01(receive(agt>thing,obj>thing):2K.@entry.@past, 1959:35)
mod:01(BA(equ>Bachelor of Arts):2X, he:2T)
obj:01(study(icl>learn(agt>thing,obj>thing)):1O.@past,literature(icl>art):25)
aoj:01(French(aoj>thing):1Y, literature(icl>art):25)
{/unl}

Output: ਉਹ ਨੇ ਟੋਕੀਯੋ ◌ਿਵਸ਼ਵਿਵਿਦਆਲਾ ◌ਿਵਚ ਦਾਖ਼ਲਾ ◌੍ ਪਾ ਯਾ ਉਹ ਬੀ . ਏ . ◌ਿਵਚ ੧੯੫੯ ਸੇਂ ਪ੍ਰਾਪਤ ◌ਿਕਤਾ ਤੇ ਫ਼ਰਾਂਸੀਸੀ ਸਾ◌ਿਹਤ ◌੍ ਪੜ੍ਹ ◌ਾ

Desired Output: ਉਨੇ ਟੋਕੀਯੋ ◌ਿਵਸ਼ਵਿਵਿਦਆਲਾ ◌ਿਵਚ ਦਾਖ਼ਲਾ ◌ਿਲਆ ◌ਿਜਥੇ ਫ਼ਰਾਂਸੀਸੀ ਸਾ◌ਿਹਤ ◌੍ ਪ◌ਿੜਹ◌ਾ ਤੇ ੧੯੫੯ ◌ਿਵਚ ਬੀ . ਏ. ਪ੍ਰਾਪਤ ਕੀਤਾ

Chapter 6: Conclusion and Future Scope

Conclusion

In this thesis, We have described in detail the various issues involved in Punjabi Generation and the ways we adopted to approach them. The work is supplemented by a java implementation of the same. Though not precise but output definitely gives a reasonable Syntax Plan and Case Marking in most cases. Imprecision of the work is reflected in handling large clausal sentences having multiple clauses. But the system is still under training phase and the accuracy of the system is purely dependent on the accuracy and the number of heuristics used. The more the number of heuristics, higher will be the precision of output.

Future Work

Future Work on this project includes the following:

- To increase the number of heuristics that are currently determining the syntax plan and making them more precise by testing there results on large corpuses.

- To extend the syntax planning to use lexical knowledge of the universal words. In the current implementation, the Syntax Plan is generated purely on the basis of relation labels. But we believe that in large complicated sentences the lexical information of the uw will have a significant effect on the syntax plan

- To make the system more linguisticaaly more strong.

- To remove disamibuigty of words by employing word net

- To improve the Case Marker rule base by adding more number of rules that uniquely determines case markers.

- To extend the corpus dictionary to include more number of words in it.

- To design Miltilingual Search Engine and an Enconverter

- To create an interface for the system.

- Extensive testing of the system.

References

[1] Bokil Hrushiklesh M.: Towards Marathi Sentence Generation from Universal Networking Language ,M.Tech Dissertation, 2002 , IIT Bombay.

[2] Dr. Saquer Abel ,Dr. Asda Abel : *UNL: A mean toBbridge Digital Divide*

[3] Durgesh Roa :MT in India:A brief survey, National Centre for Software Technical ,Mumbai

[4] Jitender Singh and Gajendra Agrawal: Information Need and Dissemination: Indian Rural Context,Indo European Systems Usability Partnership Conference, ICHI 2004 on Human Computer Interface, Bangalore, December, 2004.

[5] James Allen. *Natural Language Understanding. Pearson Education,* 2004.

[6] Kautilya Jain: Hindi Generation From UNL: Lexicon and Morphology, Mini Project Report, 2005.

[7] Monju M., Shilpa T., Smitha D., Leena G., Shachi D., P. Bhattacharyya Knowledge Extraction from Hindi Documents, International Conference on Knowledge Based Computer Systems (KBCS 2000), Mumbai, India, December, 2000.

[8] Nitin Verma and Pushpak Bhattacharyya, Automatic Lexicon Generation through Wordnet, International Conference on Global Wordnet (GWC 04), Brno, Czeck Republic, January, 2004.

[9] Pooja, V. and Yashraj, E.: *Natural Language Generation from semantc information*, B.E. Project Report, 2003, University of Mumbai.

[10] P. Bhattacharyya :*Multilingual Information Processing Using Universal Networking Language* , Indo UK Workshop on Language Engineering for South Asian Languages (LESAL), Mumbai, India, April, 2001.

[11] R..M.K.Sinha *UNL: Beyond MT*: An Indian Press ,Proceeding of LEC'02

[12] Rayner, D.: Syntax Planning, Lexicon and Morphology in Natural Language Generation, Master Thesis, 1998, IIT Bombay.

[13] Raziq Ahmad G. Saudagar. An automated generation rule for Hindi. MCA Thesis, January 1999.

[14] Ritesh Kumar Sinha: Hindi Generation : Syntax Planing and Case marking ,Mini Project Report, 2005.

[15] Rosetta, M.T. 1994. Compositional translation. Dordrecht: Kluwer.

[16] S., Parikh J. and Bhattacharyya P.: 2002,Interlingua Based English Hindi Machine Translation and Language Divergence,ournal of Machine Translation(JMT),Volume 17.

[17] Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya, Interlingua Based English Hindi Machine Translation and Language Divergence, Journal of Machine Translation (JMT), Volume 17, September.

[18] Shachi Dave and Pushpak Bhattacharyya, Knowledge Extraction from Hindi Texts, Journal of Institution of Electronic and Telecommunication Engineers, vol. 18, no. 4, July, 2001

[19] Sinha, R.M.K. and Jain, A.: Angla Hindi : An English to Hindi Machine-Aided Translation System, Machine Translation Summit IX, New Orleans, Louisiana, USA, September 23-27, 2003.

[20] The Universal Networking Language (UNL): Specifications Version 3 Edition 2, http://www.undl.org/unlsys/unl/UNL%20Specifications.htm

[21] T.Dhanabalan ,T.V. Geetha : UNL Deconverter for Tamil : International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies,December2-6,2003

[22] Uchida, Hiroshi., Zhu Meiying.,- The UNL, A Gift for a Millennium, a book published at UNU Institute of Advanced Studies.

[23] [2001.9] Uchida, Hiroshi., The Universal Networking Language Beyond Machine Translation ,"International Symposium on Language in Cyberspace" held at 26 - 27 September 2001, Seoul of Korea,Dave.

[24] W.J.Hutchins and H.L.Somers, "An Introduction to Machine Translation", Academic Press, 1999.

[25] Zhu, Meiying., Uchida, Hiroshi., UNL Center, UNDL Foundation - Universal Word and UNL Knowledge Base -invited presentation at ICUKL-2002 at 25-29 Nov. 2002, Goa of India

Paper Published /Accepted/ Communicated

1. Anjuman Chawla, Parteek Bhatia **" Introduction to UNL"** at National Conference on Computer Science and Information Technology Dept Of CSE & IT held in Baba Banda Singh Bahadur Engg College Fatehgarh Sahib on March 18-19, 2006. **[Accepted].**
2. Anjuman Chawla **" UNL Bridge Across Language Barrier"** at National Conference on Information and Emerging Technologies in Institute of Engineering and Technology, Bhaddal on March 17-18,2006. **[Published]**.
3. Parteek Bhatia, Anjuman chawla **" Punjabi Sentence Structure and Case Marking"** The 8th International Conference on Artificial Intelligence and Symbolic Computation organized Beihang University, Beijing, China, September 20-22, 2006. **[Communicated].**

Appendix A: Attribute Label Resolution Morphology

(Note: A field with a hyphen can take any value possible in that field)

Suffix	*Tense*	*Aspect*	*Mood*	*Number*	*Gender*	*Person*	*Vowel Ending*
ਦਾ ਹਾਂ	@present,	@custom	-	sg	male	1st	-
ਦੀ ਹਾਂ	@present	@custom	-	sg	female	1st	-
ਦੇ ਹਾਂ	@present	@custom	-	@pl	male	-	-
ਦੀਆਂ ਹਾਂ	@present	@custom	-	@pl	female	-	-
ਦਾ ਹੈ	@present	@custom	-	sg	male	-	-
ਦੀ ਹੈ	@present	@custom	-	sg	female	-	-
ਦਾ ਸੀ	@past	@custom	-	sg	male	-	-
ਦੇ ਸੀ	@past	@custom	-	@pl	male	-	-
ਦੀ ਸੀ	@past	@custom	-	sg	female	-	-
ਦੀਆ ਸੀ	@past	@custom	-	@pl	female	-	-
◌ਾ	@past	@complete	-	sg	male	-	a
◌ੀ	@past	@complete	-	sg	female	-	a

Suffix	*Tense*	*Aspect*	*Mood*	*Number*	*Gender*	*Person*	*Vowel Ending*
ੇ	@past	@complete	-	@pl	male	-	a
ੀਂ	@past	@complete	-	@pl	female	-	a
ਯਾ	@past	@complete	-	sg	male	-	A
ਈ	@past	@complete	-	sg	female	-	A
ਏ	@past	@complete	-	@pl	male	-	A
ਈ	@past	@complete	-	@pl	female	-	A
ਯਾ	@past	@complete	-	sg	male	-	o
ਈ	@past	@complete	-	sg	female	-	o
ਏ	@past	@complete	-	@pl	male	-	o
ਈ	@past	@complete	-	@pl	female	-	o
ੂੰਗਾ	@future	-	-	sg	male	1st	a
ੂੰਗੀ	@future	-	-	sg	female	1st	a
ਏ ਗਾ	@future	-	-	sg	male	-	a
ਏ ਂ ਗੇ	@future	-	-	@pl	male	-	a
ਏ ਗੀ	@future	-	-	sg	female	-	a
ਏ ਂ ਗੀ	@future	-	-	@pl	female	-	a

Suffix	*Tense*	*Aspect*	*Mood*	*Number*	*Gender*	*Person*	*Vowel Ending*
ਗਾ	@future	-	-	sg	male	2nd	e
ਗੇ	@future	-	-	@pl	male	2nd	e
ਗੀ	@future	-	-	sg	female	2nd	e
ਗੀ	@future	-	-	@pl	female	2nd	e
ਗਾ	@future	-	-	sg	male	3rd	e
ਗੇ	@future	-	-	@pl	male	3rd	e
ਗੀ	@future	-	-	sg	female	3rd	e
ਗੀ	@future	-	-	@pl	female	3rd	e
◌ਾਂਗਾ	@future	-	-	sg	male	1st	A
◌ਾਂਗੀ	@future	-	-	sg	female	1st	A
◌ਾਂਗਾ	@future	-	-	sg	male	1st	o
◌ਾਂਗੀ	@future	-	-	sg	female	1st	o
ਏ ਗਾ	@future	-	-	sg	male	-	A
◌ਾਂ ਗੇ	@future	-	-	@pl	male	-	A
ਏ ਗੀ	@future	-	-	sg	female	-	A
◌ਾਂ ਗਯੀ	@future	-	-	@pl	female	-	A
ਏ ਗਾ	@future	-	-	sg	male	-	o
◌ਾਂ ਗੇ	@future	-	-	@pl	male	-	o

Suffix	*Tense*	*Aspect*	*Mood*	*Number*	*Gender*	*Person*	*Vowel Ending*
ਏ ਗੀ	@future	-	-	sg	female	-	o
ਏ ਗਯੀਆਂ	@future	-	-	@pl	female	-	o

Appendix B: Rules for Punjabi Verb Morphology

VERB MORPHOLOGY for irregular forms

Only for aoj relation (with @past, @complete attributes)

Verb form	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ਹੋ	male	sg	-	-	ਇਆ
ਹੋ	male	@pl	-	-	ਏ
ਹੋ	female	sg	-	-	ਈ
ਹੋ	female	@pl	-	-	ਈਆਂ

(Verbs with @past, @complete attributes)

Verb form	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ਕਰ	male	sg	-	ਰ	ੀਤਾ
ਕਰ	male	@pl	-	ਰ	ੀਤੇ
ਕਰ	female	sg	-	ਰ	ੀਤੀ
ਕਰ	female	@pl	-	ਰ	ੀਤੀਆਂ
ਜਾ	male	sg	-	ਜਾ	ਿਗਆ
ਜਾ	male	@pl	-	ਜਾ	ਗਏ
ਜਾ	female	sg	-	ਜਾ	ਗਈ
ਜਾ	female	@pl	-	ਜਾ	ਗਈਆਂ

(Some other forms where addition/deletion takes place only at

The end of the verb form; Verbs with @past, @complete attributes)

Ending	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ੀ	male	sg	-	ੀ	ਤਾ
ੀ	male	@pl	-	ੀ	ਤੇ
ੀ	female	sg	-	-	ਤੀ
ੀ	female	@pl	-	-	ਤੀਆਂ
ੇ	male	sg	-	ੇ	ਿ ਤਾ
ੇ	male	@pl	-	ੇ	ਿ ਤੇ
ੇ	female	sg	-	ੇ	ਿ ਤੀ
ੇ	female	@pl	-	ੇ	ਿ ਤੀਆਂ

(Verbs like ਦੇ and ਲੇ with the attribute @future)

Verb form	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ੇ	male	sg	1st	ੇ	ਵਾਂਗਾ
ੇ	female	sg	1st	ੇ	ਵਾਂਗੀ

Verb form	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ੈ	male	sg	1st	ੈ	ਿ ਆ
ੈ	male	*pl*	*1st*	ੈ	ਏ

Verb form	*Gender*	*Number*	*Person*	*Deletion from the end*	*Addition*
ੈ	male	sg	1st	ੈ	ਦੀ
ੈ	female	pl	1st	ੇੈ	ਦੀਆ