Anusaaraka: An approach for MT taking insights from the Indian Grammatical Tradition

A dissertation submitted to the University of Hyderabad
for the award of the degree of

Doctor of Philosophy
in
Applied Linguistics

by

Anantpur Amba Padmanathrao
06HAPH01

Center for Applied Linguistics and Translation Studies
School of Humanities
University of Hyderabad
Hyderabad
October 2009

I hereby declare that the work embodied in this dissertation entitled **" Anusaaraka: An approach for MT taking insights from the Indian Grammatical Tradition "** is carried out by me under the supervision of Prof. G. Uma Maheshwar Rao, Center for Applied Linguistics and Translation Studies, University of Hyderabad, Hyderabad, and has not been submitted for any degree in part or in full to this university or any other university.

Anantpur Amba Padmanathrao **06HAPH01**

Date:

Place: Hyderabad

Center for Applied Linguistics and Translation Studies
University of Hyderabad

CERTIFICATE

This is to certify that **Anantpur Amba Padmanathrao** has carried out the research-work embodied in the present dissertation entittled **" Anusaaraka: An approach for MT taking insights from the Indian Grammatical Tradition "** at the University of Hyderabad. The dissertation represents her independent work and has not been submitted for any research degree of this university or any other university.

G. Uma Maheshwar Rao

Supervisor

G. Uma Maheshwar Rao

Head

CALTS

Mohan G Ramanan

Dean

School of Humanities

University of Hyderabad

Acknowledgments

Though I joined IIT Kanpur in 1991 with an intention to join Ph.D., the destiny had something else in mind, and I had to content myself with the M.Tech. degree. But my research work continued, and then I realised that it was not Ph.D. degree which I had a desire for, but it was the conducive environment and freedom to do research which I was longing for. I was always fortunate to enjoy both of these.

In my journey with the Akshar Bharati group for around two decades, I had many good opportunities to sharpen my thinking abilities and get exposed to various branches of studies such as Linguistics, Vyākaraṇa, Computational Linguistics, and Indian theories of meaning – to name a few. Chaitanyaji helped me througout this journey to seek an answer for 'what is the purpose of my life'. I do not know whether I have understood the purpose of my life, but certainly, through his life, I had a great opportunity to observe very closely, what a karma yoga is.

I had occasions to discuss various aspects of my work presented in this thesis with many scholars from different fields. I had previlage to work with linguists at the Language Technologies Research Center at IIIT, and University of Hyderabad, and with the Sanskrit scholars at the Rashtriya Sanskrit Vidyapeetha, Tirupati. Presentation of my work at various conferences, seminars, workshops and the discussions thereafter, various experiments conducted in Madhya Pradesh and Tirupati, provided good insights and directions to my work.

I express my deep sense of gratitude to Prof. K. V. Ramkrishnamacharyulu, who clarified my doubts at various occasions. I thank Dr. Shrinivas Varkhedi and Dr. Veeranarayana Pandurangi for participating in useful discussions on English Grammar. Special thanks to Dr. Shrinivas Varkhedi, who organised a conference at Rashtriya Sanskrit Vidyapeetha on 'English Grammar through Paninian Perspective'. The discussions we had on English Grammar through Pāṇinian perspective with Diptiji, and Prof. Lakshmibai from IIIT Hyderabad resulted in the lessons for Anusaaraka readers.

I thought my constant engagment with my work is causing negligance on my part

to look after my sons Achyut and Kedar. But I was wrong. They were very happy that mother is not after them asking them to study, especially in the crucial years of board examinations, giving them a lot of playing and breathing time!

Had my brothers, sister, and father - who is a constant source of inspiration for me - be not after me, it would have taken another year to complete the thesis.

Prof. Rahmat extended his helping hand by taking complete responsibility of the project on Urdu-Hindi machine Translation system, thereby reducing my load substantially.

Surekha and Raghavan built the initial version of the anusaaraka interface in Java. Later Mohan Adapa Sunil wrote the current browser version.

Finally Uma Maheshwar Rao, my friend, colleague and supervisor who in spite of his various engagements, not only went through my persistently changing manuscripts patiently, without complaining about my constant revisions, giving many useful and linguistically insightful suggestions but also importuning me for completing the work at the earliest.

My sincere gratitude to all of them.

Dedicated to my late Mother,
and late Husband Pradeep

नेहाभिक्रमनाशोऽस्ति प्रत्यवायो न विद्यते।
स्वल्पमप्यस्य धर्मस्य त्रायते महतो भयात्॥

श्रीमद्भगवद्गीता – 2.40

There is no loss of effort
nor is there any harm
(in case of production of contrary results).
Even a little of this knowledge,
even a little of this yoga,
protects one from the great fear.

Contents

List of Figures

Chapter 1

Overview

Machine Translation(MT) is among one of the first applications of computers. Since 1950, there have been several efforts [44] in this field leading to the development of several products and tools for language analysis. With the emergence of World Wide Web, enormous amount of corpora is being available which has boosted the research in Statistical techniques in the field of Natural Language Processing(NLP). The emergence of new branches of knowledge such as computational linguistics(CL), advancements in Statistical Techniques, developments of computational grammars such as Lexical Functional Grammar(LFG), Tree Adjoint Grammar(TAG), development of parsers based on these and various other formalisms such as minimalism, Link grammar, HPSG, etc., and the availability of corpora for several languages are the key factors for the growing activities in the field of NLP in general and MT in particular.

The conventional machine translation systems are fragile and do not provide a fall-back mechanism. Since the main focus of these systems is also to produce a translation, 'faithfulness' to the original text takes a back seat. With an aim to provide a faithful access to the text in other languages, anusaaraka [70] was developed. With appropriate division of load between man and machine, Kannada-Hindi anusaaraka

demonstrated that it is possible to reduce the language barrier. It was necessary for an anusaaraka reader to undergo some training on the syntactic divergences and special notation used to handle the semantic divergences between the source and the target language. Following the success of Kannada - Hindi anusaaraka system, development of 4 other anusaaraka systems among Indian Languages was undertaken jointly by IIT Kanpur and the University of Hyderabad in the early 90's. Unavailability of electronic texts in Indian languages necessary for building certain Statistical tools for language analysis, and the tremendous amount of efforts required to develop such tools manually using rule based approach put a check on further development of the anusaaraka.

With an increasing interest in MT, several efforts were undertaken by various research groups to develop language resources and tools for language analysis. Several of these resources and tools for English are available freely. With the growing need of the Indian society to have access to the English texts, and the availability of various English language resources and analysis tools, the anusaaraka group shifted its focus from building anusaarakas among Indian languages to building anusaaraka systems from English to Indian languages (with Hindi as a case study). Though various analysis tools for English were available, one can not just plug-in these components in a MT system, since these resources follow different grammar formalisms and thus do not produce an uniform output.

Thus there are two requirements – one is to provide faithful translation with reduced burden on the user and the second one is a facility to plug-in available resources thereby avoiding the re-inventing of the wheel. The goal of this thesis is to present an architecture for MT system that produces a robust and faithful output, allows plugging-in of different resources, degrades gracefully in case of failures, provides an interface to display right kind of information at right time and provides full control to the user who can navigate through various linguistic analyses and resources as per

the need.

The earlier anusaaraka system could be used only by those who have undergone some training. This puts a considerable load on the part of user. When the divergence between source and the target language is more (e.g. as in the case of English and Hindi), the load on the user is substantial. The proposed architecture reduces this burden. This architecture is planned to cater to the needs of diverse requirements such as fidelity of the translated output and naturalness of the translation.

On the face of it, this proposal may seem to be ambitious, but the insights from Indian Grammatical Tradition (IGT) and in particular the information centric analysis approach of Pāṇini provides appropriate clues leading to the required architecture.

"Pāṇini's grammar is universally admired for its insightful analysis of Sanskrit" [56]. In spite of being a grammar basically written for Sanskrit, it provides many ingenious concepts for language analysis, which are universal in nature. Pāṇinian Grammar (PG), as any other grammar formalism, provides an appropriate set of procedures to identify the relations among words in a sentence. However, the importance of PG lies in the minute observations of Pāṇini regarding the information coding in a language.

A concept of *śabdasūtra* which is influenced by the notion of *pravṛtti-nimitta* guarantees faithfulness. *Śabdasūtra* provides a core sense (or in some sense a nuclear sense) of a word. The use of *pravṛtti-nimitta* to express the meaning of English into Hindi together with the other factors such as *ākāṅkṣā*(expectancy), *yogyatā*(competancy), *tātparya*(intention) and *sannidhi*(proximity) help in the process of *śābdabodha* (understanding the meaning of a sentence).

These four factors together with the *arthabodhakatva* or *gamakatva* (ability to convey

the desired meaning) and the clues from information centric analysis by Pāṇini provided guidelines for studying the structural divergences between English and Hindi. In case machine fails to provide a translation, with the knowledge of these divergences and the *Śabdasūtra*s a user can still 'understand' the original English sentence ensuring fidelity.

The parallel processing of various modules and the changes in the order of operations guarantee the robustness and graceful degradation in case of failures. The concept of interfaces to parsers provides a plugging-in facility. A voting algorithm allows to plug-in more than one parser (or tagger) facilitating the selection of the best parse (or tag).

Thus this thesis consists of two parts – one part deals with the engineering aspect or the design of a system while the other part provides a sound scientific base for the design, ensuring that the design is based on the time tested principles thereby ruling out any possibility of ad-hoc solution.

Chapter two provides a brief history of the MT efforts within and outside India and also gives a brief summary of the new trends in this field. Typically the problems in MT arise because of the differences in two languages at various linguistic components such as syntactic, semantic, pragmatic etc. It is natural that the literature on MT discusses these problems in the light of these linguistic modules. Chapter three on the other hand, presents the problems in MT from information centric point of view. In this chapter, the process of MT is viewed as a process of decoding and encoding. Since languages code information only partially and sometimes on discontinuous strings even at far off places, decoding a string in a language can not yield the 'complete' picture. Further at the encoding point, if the two languages differ in the convention of coding, the resulting encoding may lead to catastrophe while dealing with the mapping the information in SL to TL with respect to the differences at the

level of labeling and packaging of information may crop up. In the fourth chapter, we look at the problem of MT afresh, and provide a new architecture with the desired features. We mainly discuss the proposed architecture for English - Hindi language pair. We claim that this architecture provides a better approach for MT because it is robust and transparent. Since it is based on the 'information dynamics', it also helps a developer in discovering the divergence cases more easily. Further the architecture is flexible enough to use both rule based as well as statistical modules allowing us to take an eclactic approach.

In the second part we show the appropriateness of Pāṇini's theory (chapter 5) to provide a scientific base for carrying out the information centric analysis. This study further leads to an important observation regarding the limitations of MT. The questions viz. where does a language code information?, how much information does it code?, and the manner in which it codes the information are the three aspects of the information dynamics or the parameters that are crucial in identifying the "true nature of the language". These three parameters may be used to determine the syntactic divergence between the languages. And hence we claim that any grammar which is developed with the three questions in mind: **where**, **how much** and **how** is the information coded, would be truly in Pāṇinian spirit. The insight obtained from Pāṇini's work is used to discover the reasons behind the structural divergence between English and Hindi. Chapter 6 discusses these structural divergences and concludes that all these structural divergences may be attributed to only the two missing formatives of the accusative and the yes-no question in English. It further lists three important things a Hindi reader reading an English text should tune to. They are

a) acquire a new '*vṛtti*' – the 'quazi compound' _V_

b) do away with the normal '*sannidhi*' (proximity) between a verb and its auxiliary and also between a noun and its post-position (which are integral part of Indian lan-

guages), and acquire new '*sannidhi*'s between: i) a subject and auxiliary and ii) a verb and its preposition, and
c) remember that the occupant of subject position need not have any kāraka role with the corresponding verb.

Seventh chapter explains in brief, how Dorr's divergences are handled at various levels of anusaaraka output and then mentions the three important syntactic phenomena in English viz. a) resultative constructions, b) verbs of motion specifying the manner of motion, and c) the absolutive constructions. As a consequence of these structures being absent in Hindi, there is a structural gap between English and Hindi. Hence such constructions are really problematic. Eighth chapter discusses the concept of *pravṛtti-nimitta* and how it helps in developing *śabdasūtra*s for words with various types of lexical divergences. Finally with the help of examples, we explain, how the meaning of an English sentence is understood by the user following the anusaaraka outputs at various levels leading to the *Śābdabodha*. Ninth chapter discusses the issues involved in the development of parser interfaces. The current trend is, most of the parsers produce both a phrase structure as well as dependency style output. But the output of various dependency styles do not match. We propose guidelines for producing the output in dependency style based on Pāṇinian Grammar Formalism. The tenth chapter provides future directions for taking anusaaraka further.

The central theme of the thesis is the information centric analysis. An effort is made to run this thread of information centric analysis throughout the thesis.

Chapter 2

Machine Translation: Brief History

2.1 History

The history of Machine Translation(MT) is as old as that of computers. MT is among the first computer based applications. During the second World War, the ideas from cryptography and information theory were used for deciphering the messages of enemy. In the USA, a large number of research groups sprang up to work on various MT tasks (from Russian to English), with funding from defense and intelligence establishments. In the USSR, there was a similar effort to translate from English and French to Russian. But unfortunately, in an enthusiasm and optimism during the early days, it was proclaimed that MT systems were around the corner, and that the MT systems would be capable of producing high-quality translations for general texts without any human intervention. Thus in the USA, when a committee called ALPAC was set up to evaluate the MT research, it came to the conclusion that research had not lived up to its promises. In its report in 1966, it said that basic research was needed and MT was not feasible in the near future. This made funding for MT research harder to obtain in the USA.

However, the MT research continued in Europe and Japan. After the successful com-

pletion of the TAUM-METEO system in Canada in 1977, the field regained its reputation. TAUM-METEO is a domain specific system which translates the Canadian weather forecasts from English to French. It has a very small lexicon of just 220 words and produces translations of weather reports at 98% accuracy. Around the same time other systems like Titus (English to French for textile technology), CULT (Chinese to English for Mathematics and Physics journals), etc. were also developed. In the 80s, the Japanese successfully completed a national project (Mu) on MT between English and Japanese. The European Community also undertook an ambitious project called EUROTRA covering all the languages of the European Community. It lead to the establishment of several computational linguistics groups in several European countries.

The 90s witnessed an emergence of statistical based techniques for Machine translation. CANDIDE a MT system developed by the IBM group using purely statistical techniques could stand against the Pangloss system [35] which followed the interlingua approach and the LingStat system [92] which combined the statistical and linguistic approaches. In the late 90s, an MT system Verbmobile which focused on speech to speech MT of dialogues of scheduling meetings - was developed in Germany [71].

The efforts of the MT community in the last century has started bearing fruits. The last decade has seen many usable products in the form of various online dictionaries, translation memory softwares to assist the human translators, several tools for language analysis, and of course online Machine Translation systems. The compendium of MT available from the International Association of MT [44] lists over 129 pages of commercial MT systems and computer-aided translation support tools. In addition there are several systems being developed at several Universities and research organisations and are available for free.

2.2 Approaches

It was always thought that MT requires highly sophisticated theories of linguistics in order to produce reasonable quality output. Many of the initial approaches were based on very informal and naive language analyses. New initiatives in the linguistics specifically computational linguistics led to computational grammars: Lexical Functional Grammar (LFG), Head Driven Phrase Structure Grammar (HPSG), Tree Adjoining Grammar (TAG), to name a few.

Most of the MT systems till 80s were **rule based**. That is they consisted of rules for analysis of words, rules for lexical disambiguation, rules to handle contrastive features of pairs of languages etc. Though more rules lead to sophisticated systems, they also increase the complexity and thereby maintenance of the system becomes difficult. Typically a good coverage practical MT system may have around half a million to one million head words with several thousand rules. Further, writing special rules for handling exceptional cases increases the number of rules thereby making the system clumsy.

The 90s brought a dynamism in the field of MT when the systems based on **Statistical** techniques [6] and Example Based Machine Translation [81] challenged the rule based MT systems and produced better results. The CANDIDE system [6] challenged the view that only highly sophisticated linguistic grammars / theories can deliver better quality output. MT systems which used statistical techniques to gather the rules for cross language transfer could perform equally well during the DARPA MT evaluation series in 1994. However it could not convince that statistics alone could deliver better results. In other words, with the success of CANDIDE system what one could conclude was statistical techniques can supplement the linguistic approaches. But still it remained a question as to which aspect of MT system can

be best approached by statistical methods and which by the traditional linguisticis. Since 1994 MT researchers used various hybrid models of involving statistical and linguistic techniques [52], [53], [72], [49].

Certain phenomenon can be approached better by certain techniques. For example, morphological analysers can best be built with linguistic approaches whereas POS taggers for positional languages like English may perform better by statistical methods. Linguistic approaches may perform better in places where one does not require any extra linguistic information for analysis. All the modules or phenomena that require extra linguistic information, which is very difficult to formulate, may perform well by statistical approaches. One of the major drawbacks of statistical systems is one can't fix whatever machine has 'learnt' if it has learnt something wrong. However in case of linguistics based approaches, a linguist can fix the rules so as to improve the system's performance. But to handle a linguistic based system, one needs a person with sound background in the linguistic theories concerned. On the other hand, the performance of statistical systems, normally gets better with more and more tagged corpus. Ideally, both the linguistic insights as well as statistical techniques are required and they complement each other. Linguistic insights can help in preparing a good training corpus on which one can train the statistical modules. The current statistical techniques produce the 'learnt' knowledge in terms of statistical data which is not human understandable, thereby making it impossible for a human being to fix the improper learning. It is necessary that the statistical methods produce human readable output. Such outputs not only help us to improve the learning but may also provide better insights for improvising the linguistic models as well.

Example Based Machine translation (EBMT) is another approach to MT [81]. In this approach pairs of bilingual expressions / sentences are stored in the database. The source language sentences / expressions are matched against the expressions /

sentences in the database for the best match. Naturally, the quality of translation of such a system is very high. Unfortunately, such a system needs to be addressed from the issue of scalability. For a realistic coverage, one needs a huge database which becomes unmanageable. Consequently, a combination of example based plus a rule based is a practical solution. Such a hybrid solution then can use rule based approach to cover the generalities of a language whereas it may use example based approach to handle the special cases. This helps in keeping the database manageable as well as the size of the rules in control and easy to maintain.

2.3 MT Efforts in India

India has many active groups in MT. The earliest published work in India was undertaken by Chakraborty in 1966. Research group at Thanjavur attempted Russian to Tamil translation based on the direct approach in 1985 and the first system translated simple sentences. By early 90s several research groups at different parts of the country were working on Machine Translation. A group at NCST[1] did some work on English to Hindi translation of news stories. The group at C-DAC[2] did some preliminary work on MT, but had concentrated on processing of Sanskrit. The Akshar Bharati group at IIT Kanpur focused on Indian languages and produced a system called 'anusaaraka' for accessing Kannada texts through Hindi. During the same period, another group at IIT Kanpur developed two systems 'Angla Bharati' to translate from English to Indian languages and 'Anu Bharati' to translate among Indian languages. In late 90s C-DAC started work on domain specific English-Hindi MT systems. It delivered a system that translates official letters, and is put to use at the Government offices for feedback. The IIT Kanpur Akshar Bharati group in collaboration with the Center for Applied Linguistics and Translation Studies, University of Hyderabad developed

[1]National Center for Software Technology

[2]Center for Development of Advanced Computing

'anusaaraka' systems for 5 pairs of Indian languages viz. Kannada, Telugu, Marathi, Bangla and Punjabi into Hindi.

A Fully Automatic High Quality General Purpose Machine Translation system is still a distant dream. Naturally one ends up with relaxing one or more of the constraints. For example a few groups in India tried building MT systems for highly restricted domains. Another approach is to relax the 'fully automatic' constraint allowing for either pre-editing or post-editing by humans. Some groups in India explored this approach. Both these approaches focus on translation. The very purpose of any translation system is to have access to the information coded in other languages. Anusaaraka or the language accessor is the third approach that aims at access to the original text, thereby restricting the 'high quality' to the accuracy aspect only, and thus sacrificing the ease of readability. In what follows we give a brief outline of the major efforts in India in building MT systems and language accessors.

2.3.1 MANTRA: English-Hindi Domain Specific MT

The Mantra system developed by C-DAC, translates from English to Hindi the appointment letters issued by government. It is based on Tree Adjoining Grammar and uses tree-transfer approach for translating from English to Hindi. The system is tailored to deal with its narrow subject-domain. The grammar is specially designed to accept, analyze and generate sentential constructions in official English documents. The system is deployed at different ministries [98] of government of India.

2.3.2 MaTra: English-Hindi Human Aided MT

MaTra is a technology for helping human translators to translate from English to Indian languages (currently Hindi). The main focus in this project is on the innovative use of man-machine synergy to simplify a traditionally hard problem. One of

the key features of MaTra is an intuitive structure-editor that the user can use to verify, correct and disambiguate the system's analysis of the source sentence, thus allowing a single correct translation to be produced. An advanced prototype that can handle simple sentences exists. Work is on to extend the range of sentences and to productionize the system(NCST, now CDAC-Mumbai [97]).

2.3.3 Angla Bharati System

A demo system for translation of public health campaign documents has been developed by Electronics Research and Development Center of India, Noida. The system uses the Anglabharati approach developed at IIT, Kanpur, which is based on pattern directed rule based system with context free grammar like structure for English. The system attempts to integrate example-based approach with rule-based. At the end, human being may post edit the output to correct the ill-formed sentences if any [98].

2.3.4 UNL Based MT

The Machine Translation effort at Indian Institute of Technology, Bombay is interlingua based, and uses Universal Networking Language (UNL) as an intermediate language. English/Hindi to UNL analysers and UNL to Hindi/Marathi generators are ready. The work on Marathi to UNL analyser has started. All these are rule and lexicon driven. Currently each system has about 5000 rules covering wide ranging language phenomena. The English - Hindi MT system using UNL systems employes a dictionary of concepts for Hindi which has currently around 80,000 entries in it.

2.3.5 UCSG based English-Kannada MT

At the Computer Science Department of the University of Hyderabad a Universal Clause Structure Grammar (UCSG) formalism was developed. The system was tested with the Karnataka Budget documents in English [68].

2.3.6 Tamil-Hindi and English-Tamil MT

At K B Chandrashekhar Research Unit of Anna University, work on Tamil-Hindi and English-Tamil Machine Translation systems is going on. The Tamil-Hindi system, in the beginning was developed following the anusaaraka approach. At this center, several other tools for analysis of Tamil such as FST based morphological analyser, Named Entity Recognisers, and simple parsers based on Phrase Structure Grammar were developed.

2.3.7 Example Based Machine Translation

IIIT-H in collaboration with IISC, Bangalore also worked on example based machine translation and automatic acquisition of transfer grammars and transfer lexicon.

2.3.8 Industry efforts

The IBM India Research Lab at New Delhi also initiated work on statistical MT between English and Indian languages, building on IBMs existing work on statistical MT. Another private company – Super Infosoft Pvt Ltd developed a software called Anuvadak, which is a general-purpose English-Hindi translation tool that supports post-editing. Google has also recently marked its entry in the field of MT with its English-Hindi MT tool.

All these efforts are in the direction of machine translation. Thus the 'information preservation' takes back seat. Anusaaraka systems, on the other hand, aim at providing faithful access to the original text.

2.3.9 Anusaaraka Or Language Accessor:

Anusaaraka systems or the language accessors are based on the principle of 'information preservation'. As a consequence, the anusaaraka output follows the grammar of

source language. Hence before using anusaaraka systems to access the information, the reader has to undergo a short training to read and understand the output [13]. Anusaaraka provides 'glosses' in target language for each meaningful lexical unit. There are cases where the meaning is too general or too specific. Such cases are handled by introducing some special notation to either narrow down or widen the meaning. An attempt is made to find the underlying thread(called 'śabda sūtra' or 'word formula') that connects different senses of the polysemous word. A kind of formula('sūtra' in Sanskrit) is then evolved that faithfully and unambiguously represents the connection between these different senses. Since the anusaaraka output follows the source language grammar it helps in identifying the divergences among two languages, thus acting as a stepping stone towards building a fully automatic MT system. Beta versions of five anusaaraka systems (Telugu, Kannada, Marathi, Punjabi, Bangla into Hindi) were released by Akshar Bharati group in 1998 under GPL [95]. The current focus of the group is on building English-Hindi anusaaraka.

2.4 Further Developments

The major bottleneck in developing MT systems for Indian languages was lack of lexical resources for Indian languages. The beginning of 21^{st} century saw major changes in the MT work in India. Many groups started developing various lexical resources needed for Machine Translation systems. The success of statistical methods in MT gave boost to the development of many statistical techniques for developing various lexical tools which created an enthusiasm among scientists and technologists for developing the tools for various Indian languages. This gave rise to emergence of several groups working in the area of NLP all over India. Several resource centers for developing language tools and resources were supported by the Government. However, this led to a major problem of duplication of efforts and non standardisation. Since many

Indian languages have many common features, most of the tools may be developed once and used for other Indian languages. With this view in mind, in 2006, a concept of Consortium was mooted so as to avoid the duplication of efforts and at the same time deliver the results. Five such consortia in the area of MT were formed. One for developing MT systems among the modern Indian languages, another for developing MT system from Sanskrit to Hindi, two for developing MT systems from English to Hindi and one for developing Cross Lingual Information Access system.

2.5 MT is going to stay

Given the explosion of material through World Wide Web, the demand for translation is booming and MT is the only possible answer to this demand. Most MT systems developed were meant for general purpose. Domain specific systems deliver better performance as they are tailor made to specific domains. TAUM-METEO, TITUS, CULT and PaTrans [64] are some of the successful domain specific MT systems. However, to define a domain is itself a very difficult task. We rarely come across a well defined domain. The major drawback of the domain specific systems is that they are generally not scalable to general purpose systems.

Thus we see that in the earlier years the difficulties in MT were largely underestimated, and NLP researchers were as enthusiastic as the scientists claiming to build perpetual motion machines in the late 18^{th} century. However, unlike that of the perpetual motion enthusiasts, the enthusiasm of NLP researchers did not wane. Thanks to the advancements in computational linguistics (CL), emerging statistical techniques in NLP, and advancements in computer hardware. Now we see thousands of researchers working in the emerging areas of NLP, CL, cognitive science, leading to better theories and better tools for language analysis.

Chapter 3

Problems in Machine Translation

While most theories of translation are skeptical about the possibility of translation, it is a fact that works of translations are carried out. The process of translation involves decoding of a text in one language followed by encoding it into another. The theories of translation talk about the divergence between languages at several levels. When we turn from human to machine processing of translation, various problems hitherto unexpected surface. Several problems at decoding level that are not noticed by human translators are encountered by machine, challenging the task of Machine Translation. We give below the reasons as to why decoding is not easy for a machine. Further, we discuss a number of problems at the level of cross lingual transfer of information that demand a series of resolutions at the level of encoding.

3.1 Why is decoding not easy?

In any language there is always a tension between brevity and precision. Inherent linguistic process constantly strives to avoid ambiguity and bring in precision. If one states everything explicitly then the text not only becomes very lengthy, it also leads to loss in the focus due to cluttering of information, whereas brevity helps in focusing the attention. It is a natural tendency to go for brevity leaving the resolution of

ambiguity to the context.

Look at how different languages view the activity of *cigarette smoking*. The smoking of a cigarette is a complex activity which involves many sub-activities ranging from inhaling the smoke to release of the smoke through the nostrils. Hindi expresses it either as 'sigāreta pīnā' or 'dhūmrapāna karanā'. The latter comes close to the inhaling, whereas the first one is more an idiomatic one. English speaker would use *to smoke* rather than *to inhale*. Thus we see that languages use an expression that describes one of the sub-activities of the complex activity to describe the whole complex activity. This introduces an ambiguity. Look at another example: *shelve the books* to mean *put the books on the shelf*. *Butter the bread* to mean *spread the butter on a slice of a bread*. *To butter* and *to shelf* are the verbs derived from the corresponding nouns *butter* and *shelf*, but the meanings conveyed by them are totally different. In the former case, it means *to spread* and in the later case it means *to put*. However, the semantics is so clear for a native that s/he does not find any difficulty in grasping the meaning of these verbs.

Brevity leads to overloading at the word level as well as at the structural level. In the process of interpretation, a reader uses various extra-linguistic resources such as common sense, world knowledge, language conventions, cultural background, domain specific knowledge, etc. These resources, as yet, are either not available to the computer, or even if such a knowledge is available in electronic form, it is still not clear how to use such information computationally.

The main reasons of difficulty in decoding are:

- Languages may code information only partially, and
- Languages may code information at arbitrarily long distance.

3.1.1 Languages code information only partially

In this section, we give some examples illustrating how brevity leads to overloading at various levels of coding introducing an ambiguity.

Syntactic Ambiguity

- A word may belong to more than one POS category.

  Consider the sentence

  Time flies like an arrow.

  Any normal English parser would produce several parses of this sentence, whereas human being does not even 'see' any ambiguity in this sentence unless it is pointed out. The most natural parse corresponds to the meaning - *time passes away speedily*. What are the other parses which machine makes explicit? English overloads a form to achieve brevity. For example, it is a very natural phenomenon, in English, to use nouns as verbs[1]. In the above example also, the words *time*, *flies* and *like* may have different part of speech in different contexts as shown in table 3.1. Thus there are 3*2*2 (=12) possible parses. Though

Time	flies	like	an	arrow.
N / V / A	N / V	Prep / V	Det	N

Table 3.1: ambiguity

  many of these parses are rare, one can imagine situations where each of these parses is meaningful. The implication is that we can not translate a sentences in isolation. The situation or the context "controls" the translation.

- The relations are not expressed with fine granularity.

  Consider a pair of sentences, from Sanskrit, with structural ambiguity.

[1]E.g. to butter, to shelf, etc. as seen above.

Skt: pitā putreṇa saha grāmam gacchati. (1)
gloss: father{nom.} son with village{acc.} goes. (1a)
Eng : Father goes to a village with son. (1b)

Skt: rāmaḥ dugdhena saha annam khādati. (2)
gloss: Rama{nom.} milk with rice{acc.} eats. (2a)
Eng: Rama eats rice with milk. (2b)

Here the word '*saha*' assigns a role of associative kāraka to the head words viz. *putra* and *dugdha*. Whether this kāraka role is that of kartā or karma is not marked. Thus the fact that *putreṇa saha* and *dugdhena saha* are *saha kartā* and *saha karma* respectively can be decided only by appealing to the extra-linguistic knowledge.

- Overloading the markers with more than one relation.
  English uses the same preposition to mark both a noun-noun relation as well as a noun-verb relation, and as such, English sentences are much more ambiguous than the corresponding sentences in Sanskrit or in any Indian languages which use different markers to mark these two types of relations. For example, consider an often-quoted example from English

  He saw a man on the hill with a telescope. (3)

  The problem here is of the prepositional phrase attachment. Depending on where the prepositional phrase *with a telescope* is attached - to *saw*, to *a man* or to *the hill*, the meaning changes. In one case it is a kāraka relation, and in other two cases it is modifying a noun.

Semantic Ambiguity

The concepts are language independent and are infinite. We express these concepts by the words in a specific language which are denumerably finite. Overloading a word with related meanings is unavoidable. This leads to polysemy. Further, borrowing from other languages, shift in the meaning over a period of time, etc. may sometimes introduce homonymy. Homonymy and polysemy introduce the semantic ambiguities.

Consider the sentences

Skt: saindhavam ānaya. (4)

gloss: horse/salt bring. (4a)

Eng: He played well. (5)
(a violin / cricket / in a drama?)

Eng: He went to the bank. (6)
(river bank / money bank?)

To resolve the ambiguities originating from polysemy and/or homonymy, machine needs the context, world knowledge, common sense, cultural background, etc. Moreover one can never say 'this much' extra-linguistic information is sufficient for disambiguation, since potentially there will always be exceptions. Further it is still not clear to the computational linguists[2] how to organise the world knowledge so that machines can use it effectively.

[2]There had been efforts under the CYC [96] project to organize the encyclopaedic knowledge so that machine can do reasoning. However, still the effort is far from realistic use in any Machine Translation system.

3.1.2 Information is coded at arbitrarily long distance

In the previous section we saw examples where the information for ambiguity resolution is available within a sentence. The anaphora resolution may require processing of more than one sentences. But it is not clear a priory how much text needs to be processed. Sometimes a text as much as a complete novel may have to be processed. For example, Hindi has a famous story by Guleri [36], entitled 'usane kahā thā' (s/he said), and only after reading the complete story, one gets an answer to the question 'who' said it.

Thus we see that, when a machine is translating a text, even at the level of decoding several problems crop in, which otherwise are less problematic from a human translator's point of view.

In the next section we look at the problems involved in transferring the concepts from one language to the other in the context of encoding.

3.2 Cross-lingual information transformation: Problems

Once the information in the source language is decoded, the next step is to map it appropriately into the target language. The divergences between the source language and the target language lead to the problems in this transfer. There have been studies in the divergences by many – both in the context of Machine Translation as well as in the context of Human Translation [38], [84], [30]. These divergence studies look at the differences from various aspects of encoding such as distribution and use of grammatical features viz. gender, number, person, etc., divergences at the level of syntax, semantics, pragmatics, etc.

We study the divergences from the point of view of coding of information. Words and sentences are the basic building blocks of a language. Various grammatical features get realised through them. When a language encodes information partially, the world knowledge, domain knowledge, etc. help a reader understand the text. Assuming that the same world knowledge and domain knowledge is available to the target language reader, the 'gloss' of the source language words in the target language should help a TL reader to understand the text. But this does not happen. It is because the two languages need not code the information the same way. They may differ in the *manner* they code it, *what* they code, and *where* they code. To illustrate it further, look at the following English sentence and its Hindi translation.

Eng: He is there. (7)
gloss: {3 per sg masc pron} be{pres,sg,3 per} there. (7a)

Hnd: vaha vahām̐ hai. (8)
gloss: {3 per sg pron} there be{pres,sg,3 per} (8a)

Though these two sentences are taken as translational equivalents of each other, the information content in both these sentences is not the same. The Hindi sentence is ambiguous between whether the pronoun refers to a feminine / masculine person, whie in English the pronouns He has the gender information as well. This causes a problem, when one has to translate from Hindi into English. Because, English requires some information which is not explicitly coded in Hindi either at the word or sentence level. Or in other words, a sentence level analyser is not enough to represent the information of pronouns. What is required is a pronoun reference handler. The discrepancy at the higher level of coding demands higher level of processing which in turn requires more knowledge resources, and hence is costly.

We look at the discrepancy in coding at various levels. At the word level, they may differ in labeling and packaging of concepts, at the sentence level, they may differ in the syntax, language conventions, etc. The difference in culture may get reflected at any of the levels from the word level to pragmatics. In what follows we discuss the divergences at the word and sentence level.

3.2.1 Divergence at Word Level

Sapir (1929) asserted that "the 'real world' is to a large extent unconsciously built up on the language habits of the group. [...] The worlds in which different societies live are distinct worlds, not merely the same world with different labels attached." [88] (91). The different perception and mental organisation of reality leads to various kinds of mappings between the words in one language to the words in another language. We give below some examples illustrating these points.
Given two languages, one can imagine the following possibilities:

- Lexical Gap:

  A lexical gap is an absence of a word to express a specific concept in another language. The concept may correspond to a content word or to a functional word. For example, the technical words such as 'transducer', 'electricity', 'genes', etc., the function words such as determiners in English, or the adverbial marker in Telugu. All these do not have their counterparts in Hindi. In other words, Hindi has a Lexical Gap corresponding to the English technical words, determiners in English and the adverbial marker in Telugu.

- One to One mapping:

  This is the most comfortable situation from translation point of view. Typically the proper nouns, words belonging to body parts, number words etc. fall under this category.

- One to many and many to one mappings:

  If the direction of translation is changed, then one to many mapping changes to many to one and hence we have clubbed them together. This situation arises when two languages differ in the granularity.

  The word 'uncle' in English may refer to father's (elder or younger) brother, mother's brother, or aunt's (mother's sister's or father's sister's) husband. But Hindi distinguishes between each of these relations, and has a distinct label for them. Another often quoted example may be cited from the Eskimo which distinguishes between several states of ice (formation), whereas Hindi does not even distinguish between a snow and an ice.

  One may think that many to one mapping is not at all a problem from the translation point of view, however, it is not so. English has 3 distinct third person singular pronouns corresponding to 3 genders, Hindi has only one. Now consider the following sentence

  Eng: He gave her a book. (9)

  Its Hindi translation would be

  Hnd: usane usako pustaka dī. (10)

  gloss: (S)he/it him/her/it book gave. (10a)

  Since Hindi pronoun does not mark the gender, an unambiguous sentence in English has become ambiguous in Hindi, which is not a comfortable situation.

- Overlapped regions:

  This is a very common phenomenon, especially when the two languages belong to two different families, and/or belong to two different cultures. The English verb *play* means *bajānā* when its object is a musical instrument as in *play a*

violin. It is *khelanā* when its object is a game, and it means *abhinaya karanā* when it refers to playing a role in a drama. The Hindi verbs *bajānā*, *khelanā* and *abhinaya karanā* have their own domains, and are used in other contexts as well, where the English verb *play* can not be used. For example, *to ring the bell*, Hindi uses *bajānā*.

Thus we see that these differences are either because of the absence of a concept or due to the differences at the level of labeling and packaging of concepts. The cases where there is an absence of concept is comparatively easier to handle than the cases where the domains overlap. Unless the correct shade of meaning is captured, there are more chances of such situations leading to catastrophe or even to loss in communication.

3.2.2 Divergence at Sentence Level

To study the divergence at the sentence level, we need to look into the dynamics of information coding across the languages. We seek answers to the questions like, **where** does a language code information? and what is **the manner** in which it codes the information? If the two languages differ in any of these aspects, it leads to difficulty in translation.

1. Where is the information coded?

   Two languages may code the same information in different ways. Consider the English sentence,

   Eng: Rats kill cats. (11)

   Hnd: cūhe billiyoṃ ko mārate haiṃ. (12)

   gloss: Rats cats{acc.} kill. (12a)

   English codes the information about the relation of a verb to a noun in **position**. That is the information that 'rats' are the killers and 'cats' are being killed is

coded in position. Hindi, on the other hand, requires an explicit **vibhakti** marker to code this information.

Hindi also has instances where it does not use explicit vibhakti markers as in

Hnd: rāma phala khātā hai. (13)

gloss: Rama fruit eats. (13a)

Eng: Rama eats fruits. (14)

Here the information that the fruit is eaten by Rama is not explicitly coded by any morpheme. Still a native Hindi speaker does not find it ambiguous as he uses the world knowledge to get the preferred reading. In chapter 6 we show how the information about where does a language code information helps us in understanding the kind of divergences between English and Hindi.

2. How is the information coded?

   Different languages may follow different conventions of sharing the information within or across sentences. In Sanskrit the verbal suffix 'ktvā' marks the sharing of 'kartā'. Consider the following sentence:

   Skt: rāmaḥ dugdham pītvā śālāṃ gacchati. (15)

   gloss: Rama milk having_drunk school_to goes. (15a)

   The fact that the kartā of an activity of *drinking* is same as the kartā of an activity of *going* is coded in a language convention, and not through any morpheme. In other words this information is implicit.

   The 'i' suffix in Telugu comes very close to the 'ktvā' of Sanskrit. The Sanskrit language convention of the marking of sharing of kartā has a counterpart in Telugu but it is overloaded as it also indicates the cause-effect as in the following Telugu sentence.

Telugu: pāmu karici bāludu caccipoyādu. (16)

gloss: snake having_bitten child died. (16a)

Eng: The child died because of a snake bite. (17)

Similarly consider the following English sentence

Eng: Mohan dropped the melon and burst. (18)

What is expressed through English language convention is *Mohan* burst. This phenomenon, usually known as gapping, is not present in Hindi. When the two languages differ in their conventions of information coding, i.e., one codes the information implicitly whereas the other codes it explicitly, one has to make the implicit information explicit. This is not an easy task, unless one appeals to the extra-linguistic information. Hence the cases where two languages differ in their 'language conventions' of coding information, are the cases of translation failure.

3.3 Problems at the encoding level

Translation involves understanding the original text and presenting it in another language. The presentation part involves creativity. Thus, for a given text there is no 'one specific way' of translation. What exists is a spectrum of translations and it is the translator who translates, what (s)he feels is the most appropriate one. Needs and likes of different people may demand different translations.

All these issues make the task of translation difficult and machine translation more difficult.

As we notice, the brevity necessarily implies partial encoding of the information leading to ambiguity. This calls for the use of extra-linguistic information at the level of decoding. To discover the sources of information for extra-linguistic analysis, and the additional resources that are needed may take some time. At this stage, it may not be feasible to provide such a 'Knowledge Base'. The differences between two languages in information coding make the task more complex. Assuming that the extra-linguistic knowledge (except the cultural knowledge) needed by different languages is the same, it suffices to handle the differences between the languages at various levels of language encoding.

In the next chapter, we look at the current MT architecture and provide a fresh look to the problem, resulting into a new architecture. Chapter 6 discusses the syntactic divergences between English and Hindi and provides reasons behind these divergences. In chapter 7 we see how anusaaraka provides a solution to various divergences discussed by Dorr [32]. In chapter 8 we discuss the concept of śabdasūtra to handle the divergences at the lexical level.

Chapter 4

A fresh look at the problem

4.1 Conventional Machine Translation Architecture

A typical machine translation system has a pipeline architecture as shown in figure 4.1.

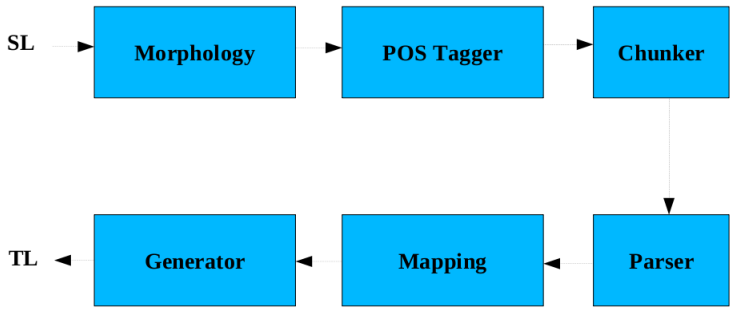


Figure 4.1: Typical Machine Translation architecture

This architecture has inbuilt 'fragility' and hence will never work with 100% accuracy. The main reason is modules like Parts Of Speech (POS) Taggers and Parsers do not give 100% performance and further the errors get cascaded leading to lower level

of performance. The best POS taggers for English give correct tags only 95-97% times. This means in one page text with around 300 words, approximately 9 to 15 words get wrong POS tags leading to wrong choice of meaning and thereby wrong translation. The best parsers for English available in the world give the first correct parse only 30-40% of the times. Thus on an average only one sentence out of three sentences is parsed correctly. The other way of measuring the performance of the parsers is in terms of relations. When a parse goes wrong, it is not that machine fails to recognise the sentence completely. It was observed that machine identifies many of the relations correctly and it is only a few relations where machine goes wrong. Hence a dependency based method for measuring the performance was suggested [59]. The best parsers in the world have around 90% performance. Translation further involves word sense disambiguation (WSD). WSD typically involves an analysis of the context, and thus heavily depends on the parsed output and the POS tags. Furthermore, since the language string does not 'express' everything explicitly, WSD also has to depend on the extra-linguistic information. This further reduces the accuracy of the translation. Since in this architecture machine makes a choice before producing any output, and the criteria for these choices are 'heuristically governed', there are chances of making mistakes by machine which are not 'human controllable'. Finally a pipeline architecture without any parallel outlets, cascades the errors and the architecture itself does not provide any fall back mechanism.

4.2 Re-visiting Translation

It is amazing that even after more than 50 years of repeated attempts, interest in MT continues to grow. The main reason behind this is the increasing NEED of the society for the translation. With the advent of World Wide Web, information is now available at the click of a button. In this age of Information, knowledge is power. Information access is a human right. As is evident today, most of the information on the web is

available only in English. In India more than 90% of the people do not understand English. This naturally will lead to a digital divide unless this information is made available to others in their mother tongue. The amount of digital information created is so enormous that it is impossible to think of getting it translated manually.

What people really need, in this era of information, is the possibility of an ACCESS to the information in other language. Anusaaraka provides an architecture to build a Machine Translation system, which in addition to providing automatic translation, also provides an ACCESS to the original text at various levels of coding such as word level, word-group level, sentence level, etc. assuring the faithfulness. We explain the concept of an ACCESSOR with an illustration of a Script Accessor followed by the concept of a Language Accessor.

4.3 An illustration of a script Accessor

A script accessor allows one to transliterate a text in one script into the other. For example, the IAST (International Alphabet for Sanskrit Transliteration) (see appendix J) is the most popular transliteration scheme used in the publications of texts related to Sanskrit, Pali, and other Modern Indian languages. This scheme uses various diacritic marks to represent the text faithfully in roman. The Graphics and Intelligence based Script Technology (GIST) is an example of a computational tool for script accessor. GIST technology provides a faithful conversion of any Brāhmi based modern Indian language text into Devanāgarī. In order to remain faithful, Devanāgarī script has been enhanced by introducing some extra graphemes (see appendix K). Figure 4.2 displays two Telugu words in Telugu script. Figure 4.3 shows the same Telugu words in an extended Devanāgarī script, transliterated by the GIST terminal. The advantage of this enhanced script is that, one can now type a Telugu text in Devanāgarī and still would be confident enough to revert back to the Telugu

script at the press of a button.

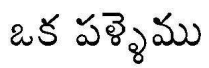


Figure 4.2: Telugu text in Telugu script

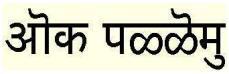


Figure 4.3: Telugu text in the extended Devanāgarī Script

The salient features of this enhanced notation are:

- faithful representation,
- reversibility,
- no loss of information,
- use of additional but minimum number of special characters,
- text in an unknown script is accessible with a little extra training.

Thus the GIST technology helps us to overcome the script barrier. The cost is: one has to put in some extra effort to learn the enhanced Devanāgarī. The effort involved in learning a few graphemes is much less as compared to the effort involved in learning a new script.

4.3.1 Difficulties in developing a script accessor

It is not always easy to develop a script accessor. Telugu and Devanāgarī being the scripts originated from the same Brāhmi script it was an easy task to develop a Telugu script accessor by enhancing the Devanāgarī script. Similarly to develop a

transliteration facility from Hindi to Telugu, one needs to enhance the Telugu script.

But in order to develop a transliteration tool from say English into Devanāgarī, it is necessary to couple the script with the pronunciation and then transliterate into an extended Devanāgarī. The heteronyms (words with same spelling but different pronunciation) are the problem sources. Such words being a few in number, the task is still doable.

When it comes to Urdu, the situation is a little more complex. Urdu and Hindi together as Hindustani, collectively form the third most populous language in the world. Both have Indian origin and have drawn from Sanskrit through Śauraseni, Apabhranśa and Khadi Bolī. The syntax of both languages is almost the same and there are many words and expressions commonly used in both the languages. Masica [63] points out that they are not even two dialects: they are exactly the same dialect used by two different communities. It is well known that Premchand wrote his stories in Urdu script and got them transcribed into Devanāgarī. The common language with common vocabulary is referred to as Hindustani that could be written in both the scripts that is Devanāgarī and Perso-Arabic. Use of two scripts for Hindustani has divided the world of Hindustani into two. Urdu uses Perso-Arabic script while Hindi uses Devanāgarī (see appendix L).

The major reasons why Urdu-Hindi transliteration is not simple is:

- Urdu does not use halanta,
- Typically Urdu does not use diacritic marks to represent short vowels,
- The semivowels *vāva*, *badī ye* and *chotī ye* are sometimes used as long vowels and sometimes they are used as consonants.

So a user reading the 'faithfully transliterated' text has to put in extra effort. In case of a semi vowel, the reader has to decide in the context whether it is a consonant or a vowel. In other cases s/he has to supply the missing vowels, and also identify the conjunct clusters and their boundaries. Figure 4.4 illustrates these difficulties. Figure 4.5 shows a glimpse of faithful transliteration leaving the burden of deciphering to the human reader who has the desired lexical knowledge.

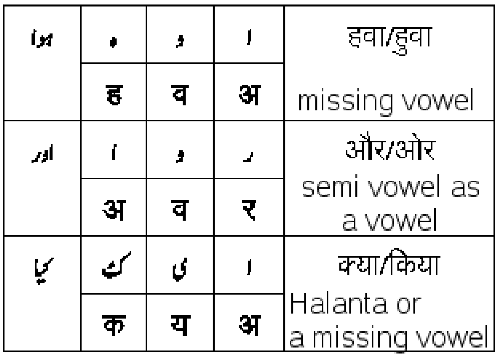


Figure 4.4: Urdu-Hindi transliteration problems

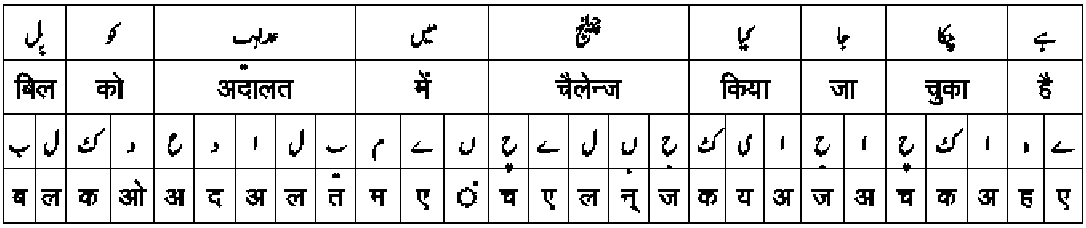


Figure 4.5: Urdu-Hindi faithful transliteration

4.4 From Script Accessor to Language Accessor:A fresh look at the problem

The task of extending the concept of 'accessor' from a 'script accessor' to a 'language accessor' is much more complex. Apart from learning a script, the learning of a language involves learning one or more of the following:

- Spelling
- Pronunciation
- Vocabulary
- Morphology
- Syntax
- ...
- New Concepts
- Culture

Just as the script accessor reduces the burden of learning a script, the language accessor is planned to reduce the burden of learning a language. Various computational tools such as morphological analysers, parsers, and disambiguation tools such as POS taggers, WS Disambiguators exist. What is needed is design of a proper architecture which provides a faithful ACCESS. Let us look at the problem of developing a language accessor afresh, taking stock of available resources and also the capabilities of the computers and humans.

1. Machines are equipped with large memory storage and high speed computing power, whereas humans are good at interpretation. So the natural suggestion is why not share the load between man and machine?

The question is how to share the load between man and machine?

2. Some of the components like morphological analyzers and dictionaries in principle can produce "accurate" information, whereas some other components like POS taggers, parsers in principle are prone to errors.

   The question is how to separate the more reliable components from the less reliable ones, and assure graceful degradation.

3. During the course of translation there exists a tension between faithfulness to the original text and naturalness in the target language making accurate translation a difficult task. Machine translation systems have a tendency to favor naturalness over the faithfulness. These systems, therefore, serve only a certain strata of people. But, there are others who would like to have an access to the "original" text (quality of faithfulness) without any "distortion" as introduced by the translation process.

   The question is, can the diverse needs of these different strata of people be addressed?

4. Finally, a machine translation system requires many linguistic tools and resources. Most of them are available for English for free download on the Internet under General Public License (GPL). It is natural to make use of these resources rather than reinventing the wheel. In fact there are more than one parser, POS tagger and morphological analyzer for English available under GPL.

   How should the system be designed so that one can plug-in different language tools such as POS taggers, parsers, morphological analyzers, etc.?

4.4.1 Earlier efforts

The earlier efforts of building Language Accessors (anusaarakas) among the Indian languages have answered the first and the third questions. The first anusaaraka system

was built in the early 90's from Kannada to Hindi [70]. The claim of the anusaaraka was that it is possible to overcome the language barrier in India using anusaaraka. The Kannada-Hindi anusaaraka demonstrated how one can take advantage of the relative strengths of the computer and the human reader to build a practical system. Special notational devices were devised to bridge the gaps between Kannada and Hindi at the word as well as sentence level. This introduced some amount of burden on the reader. But this at the same time provided 'faithful' image of the original text into the target language, which any serious reader would like to have. For the casual readers, anusaaraka provided a post-editing tool, with which a Hindi editor can edit the text semi-automatically, and generate a text which sounds natural to a target language reader. Possibility of using custom-made dictionary allowed a user to produce output of his choice.

Four more anusaarakas from four Indian languages viz. Telugu, Marāṭhī, Punjabī and Bānglā into Hindi were built in the next few years. The alpha versions of these four systems demonstrated the feasibility of the extension of different pairs of languages. The output of these anusaarakas 'closely followed' the source language constructions and the source language semantics, thereby influencing the output. The more the source language is closer to the target language, lesser is the effort a human reader has to put in understanding the text and the more close is the output to the target language.

The lack of sufficient corpus, lack of bilingual as well as monolingual linguistic resources necessary to build computational tools and lack of motivation or necessity to access texts in other Indian languages put a temporary check on the further development of these anusaarakas among Indian languages.

4.4.2 Expanding the horizons

The advent of World Wide Web, availability of enormous amount of information in English, availability of several computational Linguistic resources for English under General Public License (GPL) opened up new vistas for the anusaaraka group to develop the English-Hindi anusaaraka. The second and fourth questions raised in sec 4.4 thus become very relevant in the context of English-Hindi anusaaraka. With the availability of various computational tools for analysis of English and also libraries for developing GUI easily, the following guidelines emerged naturally influencing largely the architecture of anusaaraka.

4.5 Anusaaraka Guidelines for developing a MT system

- Make complete information available to the user but, do not clutter the scene.
- Seperate the resources that can be made, in principle, reliable from those that are, inherently unreliable. Mention explicitly the degree of reliability.
- Provide alternative means to get the information in case it is not already made available.
- Do not reinvent the wheel.
- Use existing resources and tools.

How do we achieve this?

- Make complete information available to the user but, do not clutter the scene.
  — Ensure Substitutivity and Reversibility,
  — Hiding Mechanism/User Interface.

- Seperate the resources that are, in principle, reliable from those that are, inherently unreliable. Mention the degree of reliability explicitly.
  —– Run various modules in parallel, and show the output at various levels in descending order of reliability.

- provide alternative means to get the information
  —— Provide online help,
  —— Develop Human readable algorithms first and then implement whatever is possible with today's technology and resources.

- Do not reinvent the wheel
  —— Resort to GPL.

- Use existing resources and tools
  — Develop suitable interfaces to 'plug and test' different tools,
  — In case several tools are available for a particular task, use voting to select the best output.

4.6 Architecture of the Anusaaraka system

The Anusaaraka system has two major components.

1. anusaaraka engine, and

2. User-cum-developer interface.

Anusaaraka engine is the main engine of anusaaraka. This engine produces an output in different layers making the process of Machine Translation transparent to the user. The architecture of anusaaraka system is shown in Figure 4.6. This architecture differs from the conventional architecture in three major ways—

1. Various language analysing modules for Source language such as morphological analyser, POS tagger, Chunker, Parser, etc. are run in parallel. These modules,

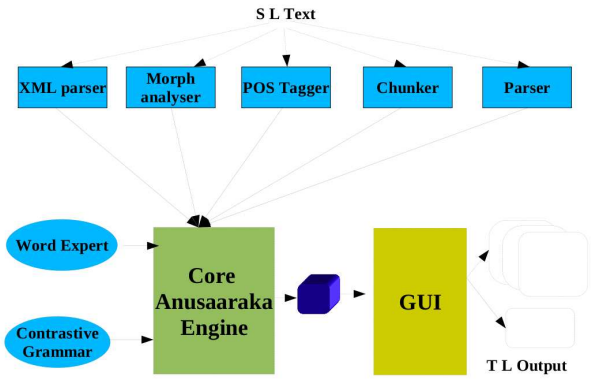


Figure 4.6: Anusaaraka Engine plus GUI

along with the knowledge of the contrastive features between English and Hindi and handcrafted rules for WSD, are used for WSD and determining the phrase boundaries, identifying the phrasal head, etc. The image of source language is shown at various levels of information encoding such as word level, phrase level and sentence level.

2. A graphical user interface has been developed to display the spectrum of outputs. The user has flexibility to adjust the output as per his/her needs.

3. Special 'interfaces', which act as 'glue', have been developed for different parsers, which allow plugging in of different parsers thereby providing modularity.

4.7 Anusaaraka engine

The current anusaaraka engine has five major modules viz.

1. Word Level Substitution,

2. Word Grouping,

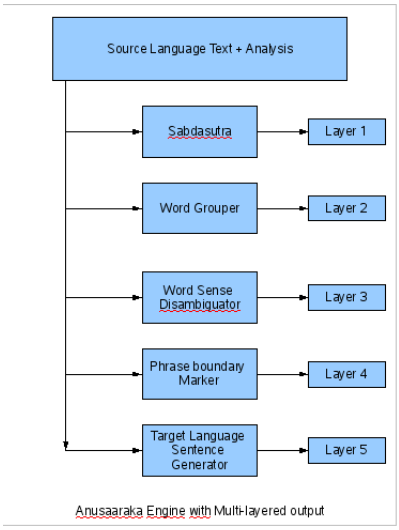


Figure 4.7: The core anusaaraka

3. Word Sense Disambiguation,

4. Phrase Boundary marker,

5. Target Language word order generator.

Each of the above five modules is described below in detail. A justification of how these modifications are appropriate in answering the questions raised in section 4.4 is presented. The concepts involved behind each of these modules are language independent. For the ease of putting across the concepts, we have used English-Hindi language pair for which the system has been tested.

4.7.1 Word Level Substitution

Every word is split into two parts – a base and a suffix. This module provides a 'gloss' of each source language word (i.e. the base and its suffix) into the target language. To provide a 'gloss' of polysemous words without compromising on the 'faithfulness' is a challenge. We discuss the solution anusaaraka offers to handle the polysemy.

4.7.2 Concept of Padasutra

By looking at various usages of any polysemous word, one may observe that these polysemous words have a 'core meaning' and other meanings are often natural extensions of this core meaning. In anusaaraka an attempt is made to relate all these meanings and show their relationship by means of a formula. This formula is termed *Padasūtra*. (The concept of Padasūtra is based on the concept of 'pravṛtti-nimitta'[1] from Indian Grammatical Tradition.) The word padasūtra itself has two meanings:

1. a thread connecting different senses of a pada,
2. a formula specifying the meanings of a pada.

Here is an example of Padasūtra:
The English word *leave* as a noun means *chutti* in Hindi, and as a verb its Hindi meaning is *choḍa*. We see that the two meanings are related. *Leave* basically is a verb and the corresponding noun gets derived from this verb. Hence the Padasūtra for *leave* is

leave: choḍa[>chuṭṭi]

Here 'a[>b]' stands[2] for 'b is derived from a'.

The concept of padasūtra takes care of two basic principles of anusaaraka viz. **substitutivity** and **reversibility** [70]. Substitutivity ensures that all the meanings of the source language word have been taken care of. Or in other words, irrespective of the context, one can substitute the padasūtra for a given word. The reversibility ensures that the substitution does not create ambiguity. At this level, machine takes the load

[1]Refer to the chapter 8 for more details.

[2]for the notation used in the padasūtras, refer to the appendix B

of morphological analysis and the lexical transfer. And the load of interpreting the text by selecting appropriate choices taking into account the context of the source language lies with the reader. Thus, by division of workload and adoption of the concept of 'padasūtra—word formula', the approach guarantees that the first level output is 'faithful' to the original and also acts as a 'safety net' when later modules fail. Table 4.1 shows an example of the output after substituting the word formulae.

He	kept	the	book	on	the	table.
vaha{pu.}	rakha_{ed/en}	the	pustaka	on	the	ṭebala.

Table 4.1: padasutra-output

At this level some of the English words like function words, articles, etc. are not substituted. The reason being they are either highly ambiguous, or there is a lexical/conceptual gap in Hindi corresponding to the English words (e.g. articles), or substituting them may lead to a catastrophe. For example, if prepositions are substituted at this stage, it may result in a catastrophe as shown in table 4.2.

He	kept	the	book	on	the	table.
vaha{pu.}	rakha_{0/ed/en}		pustaka	para		ṭebala.

Table 4.2: Catastrophe

It is likely, just by looking at the Hindi layer - the proximity of *pustaka* and *para*, that one may interpret it as 'He kept the table on the book'.

4.7.3 Training Component

To understand the output produced in this manner, a human being needs some training. The training presents English grammar through the Pāṇinian view. Chapter 6 describes the methodology followed to arrive at the contrastive English Grammar

a Hindi speaker has to acquire in order to follow the anusaaraka output faithfully. Thus, if a user is willing to put in some effort, s/he has complete access to the original text. The effort required here is that of making correct choices based on the common sense, world knowledge, etc. This layer ensures that the layer produces an output, which is a 'rough[3]' translation that systematically differs from Hindi. Since the output is generated following certain principles, the chances of getting mislead are less. Theoretically, the output at this layer is reversible.

4.7.4 Word Grouping

Sometimes a group of words trigger a new meaning. This is typically the case with compounds and idioms. For example, the group of words 'the big cat' refers to 'animals belonging to the cat family who are able to roar and live in the wild'. While translating into Indian languages naturally one can not get the correct sense if it is translated as 'badī billī'. Similarly, the verb groups – sequence of auxiliaries followed by a main verb need to be translated as one unit. For example, the verb group 'are going' needs to be translated as one unit, as *jā rahā hai*{*ba.*}, and not individually separately.

It is possible that in some cases more than one ways of groupings is possible, as illustrated below.
They (are playing) cards.
These are (playing cards).
In such cases, the system should produce both the answers. Of course, in such cases, we know that the words in the proximity provide the clues. But, in principle, the information may not be available in the close proximity. The later modules use the parser and may produce the correct answer. But in case a parser fails, then this layer

[3] *rough* here means rough as in *rough journey*, where the journey takes you to the destination, though it is uncomfortable

provides possible answers to the users, from which the user can select the correct one. Each of these layers ensures that there is a fallback mechanism in case the later modules fail.

4.7.5 Word Sense Disambiguation (WSD)

The next module in anusaaraka is the Word Sense Disambiguation.
The WSD task may be split into two classes:

1. WSD across POS
2. WSD within POS

The POS taggers can help in WSD when the ambiguity is across POSs.
For example: Consider the two sentences

1. He chairs the session.
2. The chairs in this room are comfortable.

In the first sentence, the word 'chairs' is a verb, and in the second sentence it is a noun. The POS taggers mark the words with appropriate POS tags. In the above examples. if the words are marked with correct POS tags, then the disambiguation is done. The POS taggers use certain heuristic rules, and hence may sometimes go wrong. The reported performances of these POS taggers vary between 95% to 97%. Nevertheless they are still useful, since they reduce the search space for meanings substantially.

Disambiguation in the case of polysemous words requires disambiguation rules. It is not an easy task to frame such rules. It is the context, which plays a crucial role in disambiguation. The context may be

1. the words in proximity, or

2. other words in a sentence that are related to the word to be disambiguated.

The question is how can such rules be made efficiently? To frame disambiguation rules manually would require hundreds of man-years. Is it possible to use machines to automate this process?

The WASP workbench [54] is the best example of how, with the help of a small seed data, machines can learn from the corpus and produce disambiguation rules . Anusaaraka used the WASP workbench to semi-automatically generate these disambiguation rules. The WASP generated rules though are human readable, many a times is just a list of several cases together. As such these rules lack generalisation. Advantage of WASP generated rules is, it is very easy to add, delete and modify the rules. But lack of generalisation explodes the number of rules to a few hundreds for a single word. This makes it un-manageable for a human.

The output produced at this stage is irreversible, since machine makes choices based on heuristics.

4.7.6 Phrase Boundary Marker

A phrase boundary marker is essential to decide the position of the head of the phrase. When one language is head-initial, and the other one is head-final, as in the case of English and Hindi, it is necessary to *move* the head to the appropriate position. This requires a phrase boundary marker. We provide below an example from English-Hindi pair.

While moving the prepositions from their English positions to the appropriate Hindi positions, a record of their movements is stored, making the transformations reversible. This layer marks the movement by an arrow (->). For example, ->+2

in table 4.3 indicates that the postposition *on* is moved to +2 positions in Hindi.

He	kept	the	book	on	the	table.
vaha{pu.}	rakhā		pustaka	->+2		ṭebala_para.

Table 4.3: preposition movement

4.7.7 Target Language Word Order Generation

In this step, proper word order for the TL output is generated. We discuss this module with an example of English-Hindi. Hindi is a free word order language. Therefore, even the anusaaraka output at previous layer makes sense to the Hindi reader. However, this output not being natural in Hindi, may not be enjoyed as much as the one with natural Hindi order. Additionally, it would not be treated as a translation. Therefore in this module an attempt is to generate the correct Hindi word order.

The basic structure of English is SVO. Though Hindi is a relatively free word order language, it is a statistical fact that Hindi has SOV word order. The rules for transforming the English word order into Hindi word order are being worked out. In order to go back, if needed, to English words and English word order, we also maintain a mapping between English word order and Hindi word order.

4.8 Interface for different linguistic tools

The second major contribution of this architecture is the concept of 'interface'. Machine translation requires language resources such as POS taggers, morphological analyzers, and parsers. More than one kind of each of these tools exist. Hence, it is wise to use these tools. However, there are problems.

For example several parsers for English exists on the internet. Link parser [85], Stanford parser [61], Enju Parser [94], Minipar [59] are some of them.

1. These parsers do not have satisfactory performance. At the most 40% of the time, the first parse is the correct parse. Parse of a sentence tells how the words are related to each other. 90% of such relations in any parse are typically correct.

2. Each of these parsers is based on different grammatical formalism. Hence, the output they produce is also influenced by the theoretical considerations of that grammar formalism.

3. Since the output format for different parsers is different, it is not possible to remove one parser and plug in the other one.

4. One needs a trained manpower to interpret the output produced by these parsers, and to improve the performance of these parsers.

As a machine translation system developer who is interested in the "usable" product one would like to plug-in different parsers and watch the performance. May be one would like to use combinations of them, or may like to do voting among different parsers, and choose the best parse out of them.

The question then is how to achieve it?

It is not enough to have the programs modular. The parser itself is an independent module. What is required is a plug-in facility for different parsers. This is possible provided all the parsers produce an output in some common format. Hence, interfaces are necessary to map the output of parsers to an intermediate form as illustrated in figure 4.8. This interface is based on Pāṇinian Grammar. We discuss the theoretical

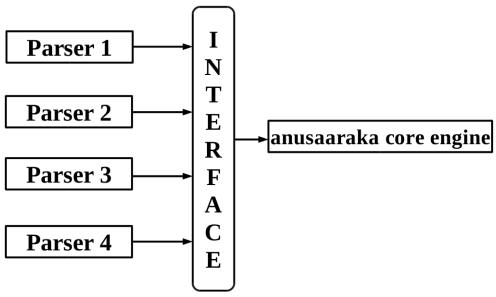


Figure 4.8: The parser interface

details of this interface in chapter 9.

We also need a voting program, which decides the best parse among several parses. Such a module then can help one to choose the best parse for a given type of sentence, or may even 'generate' altogether a new parse taking the best relations of various parsed outputs. Since no two parsers agree in their output schemes, we tested the voting algorithm on POS taggers, all of which use the same tagset though they differ on formatting details. Figures 4.9 shows the architecture of such a voting machine, and appendix A gives a sample output of such a voting algorithm, run on 5 POS taggers.

4.9 Anusaaraka output and the user interface

Typically languages differ at various levels of encoding viz. morphology, syntax, semantics, etc. Anusaaraka aims at providing 'faithful' image of the source language text into the target language. Since the incommensurability between the languages is at various levels, it is not enough to present the 'gloss' or śabdasūtra. What is required is a multi dimensional view of the coding – the dimensions referring to various

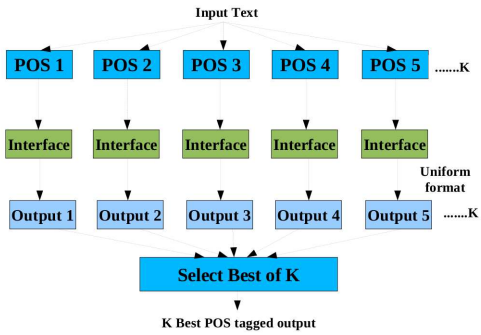


Figure 4.9: K Best POS tagger

levels of encoding. To give an analogy, an architect provides various 2-D views of a 3-D building such as plan, elevation, side view etc., similarly we in anusaaraka provide 'gloss' of the source language into TL from 'several views'. The various views are: coding at the word level, word group level encoding, coding at the level of sentence structure, coding at the phrasal level, etc.

We discuss below the requirements of a GUI for anusaaraka with reference to various kinds of information that this interface is supposed to display, and finally with the desired features.

4.9.1 Requirements of an Anusaaraka GUI

- **User profile**

  The anusaaraka GUI should cater to the diverse needs of users. The anusaaraka users may broadly be classified into

  – Users who are comfortable in English but face difficulty in constructions such as

  `Before holding a person responsible for a crime and according`

`punishment, the motive behind the action must be determined.`

For such users, help on words such as *according* should be provided. These users are aware of other regular usages such as 'according to', but may not know the meaning of verb 'accord'.

- Users who are comfortable with simple sentences but face problems with complex verb formations such as 'let out', 'make out', 'got off', etc.
  The system should provide help on such complex verb formations, compound words, etc.
- Users with poor knowledge of English with respect to even common syntactic phenomena.
  Such users need online lessons of English grammar, with an intelligent user interface providing the necessary help.
- Users who are very poor in English and also weak in analytic skills.
  For such users, a layer with Word Sense Disambiguation output and also a phrase boundary marker should provide the necessary help.
- Users who want only ready made translations.
  The system should have an access to outputs of various other MT systems, so that in case one MT system fails, other can provide the gist.

- **Features**

  Users are of two types – intuitive and sensible. The intuitive users would like to understand the system from a holistic view and then would like to take control and operate themselves. The sensible users on the other hand would like the features 'up front' or 'in their face'. Thus the toolbars, dialog boxes etc. are more appropriate for such users. The sensible users, typically outnumber the intuitive users. Also since a tool such as anusaaraka is used by the user only when in need, even intuitive users may prefer to use the dialog box and tool

bars rather than having full control over the system. The developers , on the other hand, would like to have full control over the system.

- **Shortcuts**

  A good interface should also provide 'intuitive' shortcuts to avoid the pressing of combinations in a sequence, and deep navigation.

- **Help**

  A help on the usage of interface should be handy.

- **Aesthetics**

  The interface design, the layout, the fonts used, the color combinations should give an aesthetic feeling and relish the user while using the interface.

4.9.2 Contents of the Anusaaraka Interface

The content that needs to be displayed is

- Input text in SL,
- Translated output in TL,
- Intermediate outputs consisting of
  - Notes on the śabdasūtra,
  - TL meaning after possible word groupings,
  - POS tags,
  - Chunk Boundaries,
  - Parser outputs,
  - Selection of meaning after WSD.
- Outputs of various other MT systems,

- Help on divergence between SL and TL,
- Help on specific but difficult SL constructions.

4.9.3 Anusaaraka Interface

Browser based user interface has been developed to display the outputs produced by different layers of anusaaraka engine. The user interface provides a flexibility to control the display. Various dimensions of the user interface are provided by introducing the following:

- frames
- foldable table
- links leading to pop-up windows
- tooltips
- color codes

Figure 4.10 shows the snapshot of sample English-Hindi anusaaraka output. The output is shown in three frames. The top left frame shows the original English text. The bottom frame gives the translation as any machine translations system would give. The top right frame gives a layered output showing the outputs after each of the tasks outlined above. The pop up window helps the user to hide layers in the right top frame.

A short description of each of the layers in the figure 4.10 is given below.
Frame1

1. Row 1: Original English sentence
2. Row 2: Word level substitution

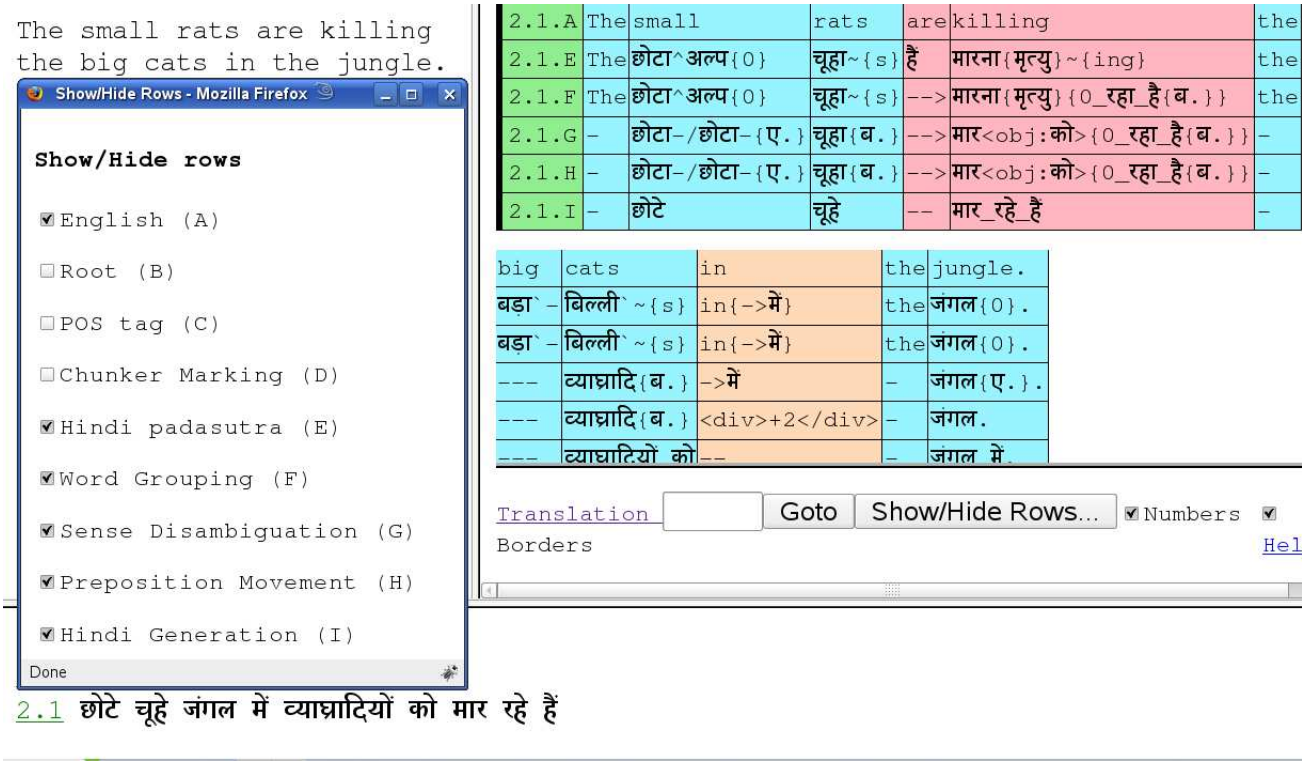


Figure 4.10: Snapshot of sample anusaaraka output

(a) It is the least fragile layer.

(b) Contains Hindi Padasutra (word formula) for each English word.

E.g. small -> *chotā^alpa*

rats -> *cūhā{s}*

3. Row 3: Word Grouping

The group of words which as a group gives a new meaning are grouped together.

For example in the above sentence,

are + ing = *0_rahā_hai{ba.}*

4. Row 4: Word Sense Disambiguation

Attempts to select the appropriate sense according to the context.

For example, the big cats -> *vyāghrādi*

5. Row 5: Phrase Boundary Marker

   Different colors mark the phrase boundaries. The movement of prepositions is marked. For example, the Hindi counterpart of 'in' is to be moved by '+2' words.

6. Row 6: Hindi Generation at word group level

   The prepositions are moved to their correct Hindi positions. Proper forms of the words are generated, taking into account the gender, number, person agreement rules of the Hindi language.
   E.g. '->*meṁ* – *jangala*' is changed to '— — *jangala+meṁ*'.

   cūhā{ba.} -> cūhe, etc.

Frame2 : Hindi Translation

1. Proper Hindi sentence is generated.

4.10 Anusaaraka: A better approach for Machine Translation

We claim that anusaaraka is a **better** approach for Machine Translation because it is

1. Robust: It always produces the output. If the machine fails at higher levels, which are in principle fragile, lower level outputs are still available to the user. However to understand the lower level outputs, some training is required.

2. Transparent: The output at different levels makes the whole process of machine Translation transparent to the user. This opens up a new opportunity

for the persons having an aptitude for language analysis to contribute to the Machine Translation efforts even without any formal training in computational linguistics, or NLP.

Table 4.4 describes the differences between anusaaraka and any traditional Machine Translation system.

Feature	Typical MT System	Anusaaraka
Goal	Aims to produce Natural translation. In case of failures produce 'rough' [4] translation.	To provide access to the Source Language text. The output may be 'rough' [5]
Unit of input	Currently, independent sentences	a complete XML document
System Components	Morph analysers, POS taggers, Parsers, Sense disambiguation modules, Generator	Same as in MT Plus Anusaaraka User Interface
Sequence of Operations	Outputs are cascaded. So errors too get cascaded	The basic tasks are processed independently. Price paid: Duplication of effort
Transparency	Processing is not transparent to the end-user	Processing is transparent to the end-user
Access	User has access only to the final output	User has access to the output at each level

[4] what is 'rough' is not well defined.

[5] Here rough is as in the sense of 'rough journey' where you are taken to the destination, but the journey is not comfortable.

Guidelines for Linguists	No specific guidelines	First write an algorithm for 'Human beings' and not necessarily for 'computers'!
Principle	Ad-hoc	"Information Dynamics"
Approaches	1. EBMT 2. Rule based 3. Statistical 4. Hybrid	Eclectic: Choose the best of each of these approaches. Use best of the available resources under GPL.

Consequences	1. Later modules are affected by the errors of the previous modules 2. Rough is not well defined. Hence users may get mislead. 3. User can not participate in the development process 4. Linguists end up in reinventing the wheel again and again.	1. Parallel processing ensures that different modules do not interfere. 2. Well defined 'Roughness'. Theoretically no chances of user getting mislead. 3. User can participate in the development activity. 4. Linguist prepares data only once.

Chapter 5

Information Coding in languages: Some insights from Pāṇinian Studies

5.1 Introduction

India records 2500 years of rich heritage in linguistic studies. Out of the six *vedāṅga*s (fields of studies necessary to study the vedas) viz. *śikṣā, vyākaraṇa, chandas, nirukta, jyotiṣa* and *kalpa*, the first four are concerned with language studies. *Śikṣā* deals with pronunciation, *vyākaraṇa* with grammatical aspects, *chandas* with prosody and *nirukta* with etymology. Among all these the importance of *vyākaraṇa* is long recognised and is evident from the enormous literature on vyākaraṇa. It has a major role to play in understanding how a language communicates thoughts from one human being to the other.

Pāṇini consolidated all the earlier descriptions for Sanskrit and presented a concise and almost exhaustive descriptive coverage of the then prevalant Sanskrit language. This grammar is in the form of aphorisms – around 4000 divided into 8 chapters of 4

sections each. "Pāṇini's grammar is universally admired for its insightful analysis of Sanskrit" [56]. In spite of being basically a description of Sanskrit, it provides many ingenious concepts for language analysis, which are universal in nature.

"The goal of Pāṇinian enterprise is to construct a theory of human communication using natural language" [9](p 59). No doubt, Pāṇinian Grammar (PG), as any other grammar formalism would give, gives a very good theory to identify the relations among words in a sentence. However the importance of PG lies not in an exhaustive description of a language but in the minute observations regarding the information coding in a language.

In the next section we establish our claim that Pāṇini was aware of various means a language engages itself to code an information. This is evident from the way he analysed Sanskrit and also from the way he framed the sūtras. We cite an example from the *Māheśvarasūtra*s. The third section discusses three sūtras from the *Aṣṭādhyāyī* and show how they answer important questions related to information coding.

5.2 Various means of encoding information: An illustration from Māheśvarasūtras

The *Māheśvarasūtras* form an integral part of the *Aṣṭādhyāyī*. It consists of 14 sūtras. Each sūtra has one or more phonological segments terminated by a marker (*anubandha* or *it*). Pāṇini has used around 42 different subsets of phonemes in the *Aṣṭādhyāyī*. The *Māheśvarasūtra*s are a linear arrangement of these 42 partially ordered sets (known as *pratyāhāra*s) with markers placed in between (at the end of each sūtra) indicating different set boundaries. The linear arrangement with markers helps one to obtain the 42 sets by a mechanical procedure thereby facilitating an easy memorisation of these sets. Kiparsky (1991) and Petersen (2004) have given

respectively the linguistic insight and the mathematical proof of the optimality of the *Māheśvarasūtras* with respect to the placement of the markers as well as the number of markers. Petersen has elegantly shown why the repetition of h in the sūtras is necessary and that the choice of h is optimal.

Pāṇini used the same consonant $\dot{N}$ as an anubandha at two different places in the *Māheśvarasūtra*s. There has not been any satisfactory explanation of the reasons behind the repetition of Ṇ. Patañjali asks, was there a dearth of phonemes that Pāṇini used the same consonant twice introducing an un-necessary ambiguity? The commentaries by Patañjali and Bhartṛhari provided an important insight into the problem from information coding point of view. Let us look at the case in detail and see what the commentators have to say. Here are the first 6 *Māheśvarasūtra*s with repeated $\dot{N}$.

$$a\ i\ u\ \dot{N}$$
$$ṛ\ ḷ\ K$$
$$e\ o\ \dot{N}$$
$$ai\ au\ C$$
$$h\ y\ v\ r\ T$$
$$l\ \dot{N}$$

This makes the pratyāhāra $a\dot{N}$ and $i\dot{N}$ ambiguous since the pratyāhāra $a\dot{N}$ may refer to $\{a\ i\ u\}$ or $\{a\ i\ u\ ṛ\ ḷ\ e\ o\ ai\ au\ h\ y\ v\ r\ l\}$, and the $i\dot{N}$ may refer to $\{i\ u\}$ or $\{i\ u\ ṛ\ ḷ\ e\ o\ ai\ au\ h\ y\ v\ r\ l\}$. Patañjali examines all the sūtras that use $a\dot{N}$ and $i\dot{N}$ and finally concludes that in each of these cases one can resolve the ambiguity. Bhartṛhari's commentary on the *Mahābhāṣya*, the *Dīpikā*, is worth mentioning. Bhartṛhari [1](p.90) observes that *sāmarthya* (ability to convey a specific meaning), *prasiddhi* (frequency of usage), *liṅga* (indicator) and *lāghava* (economy) are the deciding factors for resolv-

ing the ambiguity arising because of the repetition of *Ṇ*.[1]

The *Aṣṭādhyayī* has five sūtras that use the pratyāhāra *aṆ*. They are

ḍhralope pūrvasya dīrgho'ṇaḥ (6.3.110)[2]

ke'ṇaḥ (*aṅgasya hrasvaḥ*) (7.4.13)[3]

aṇo'pragṛhyasyānunāsikaḥ (*vā*) (8.4.56)

uraṇ raparaḥ (1.1.50)

aṇudit savarṇasya cāpratyayaḥ (1.1.68)

In what follows we show how in each of these cases ambiguity can be resolved.

5.2.1 *Sāmarthya* (ability to convey proper meaning)

The first 3 cases viz.

ḍhralope pūrvasya dīrgho'ṇaḥ (6.3.110)

ke'ṇaḥ (*aṅgasya hrasvaḥ*) (7.4.13)

and

aṇo'pragṛhyasyānunāsikaḥ (*vā*) (8.4.56)

contain the words *hrasva*, *dīrgha* and *pragṛhya*. These terms refer only to vowels. In other words, there are no cases where they qualify any of the phonemes from the set $\{h\ y\ v\ r\ l\}$. Therefore Patañjali argues that if in these three sūtras *Ṇ* were to refer to the 2nd *Ṇ* in the pratyāhāra sūtras, it would have been sufficient to use the pratyāhāra *aC* which refers to the set of vowels. Since the pratyāhāra *aC* is already in

[1] *ayam ṇakāro dvir anubadhyate. atra prakaraṇe ṣaṭprakārāḥ upakṣiptaḥ – āsattiḥ vyāptiḥ sāmarthyam prasiddhir liṅgam lāghavam iti.*

[2] In references to the *Aṣṭādhyayī*, numbers separated by periods are to chapter (*adhyāya*), section (*pāda*), and sūtra, respectively, for example, 6.3.110 indicates the 110^{th} sūtra in the 3^{rd} pāda of the 6^{th} adhyāya.

[3] Words in brackets are understood to recur from earlier sūtras. This recurrence is termed as *anuvṛtti*

use and hence does not lead to the introduction of an additional pratyāhāra, economy (*lāghava*) would be achieved. In fact, additional economy would be achieved at sūtra level, he argues, because even *aC* would not have to be mentioned, being the default case. The fact that Pāṇini has mentioned *aṆ*, therefore implies that he meant the first *aṆ* and the pratyāhāra referring to the smaller set {*a i u*} and not the bigger one. Thus it is the words *hrasva*, *dīrgha*, and *pragṛhya* in the context that determine the meaning of *aṆ*. Bhartṛhari terms this as *sāmarthya*, the ability of a particular meaning to express itself (in a particular context).

5.2.2 *Prasiddhi* (frequency of usage)

In the next sūtra *ur aṆ raparaḥ* (1.1.50), the possibility of the 2nd *Ṇ* is ruled out on the basis of unavailibility of any example which involves bigger set {*a i u ṛ ḷ e o ai au h y v r l*}. Patañjali discusses two examples in his commentary and he points out that either the effect of the rule is nullified by another sūtra, or the application of this sūtra leads to redundancy in some other sūtra, which is undesirable. Hence he concludes that if Pāṇini meant the 2nd Ṇ, he could have used the smaller pratyāhāra *aC*. Since Pāṇini used *aṆ*, in the absence of any other clue for decision, Patañjali concludes that *Ṇ* in this sūtra is the 1st one and not the 2nd one (because in all the previous sūtras involving *aṆ*, it is the 1st *aṆ* that is being used). According to Bhartṛhari, it is the *prasiddhi* (frequency of usage) which is the deciding factor in this sūtra.

5.2.3 *Liṅga* (marker)

The 5th sūtra that uses *aṆ* is

aṇudit savarṇasya cāpratyayaḥ (1.1.68)

From this sūtra alone it is not obvious which *aṆ* is meant. There is another sūtra *ur ṛt* (7.4.7) which says *ṛ* becomes *ṛt*. The *t* in ṛt makes ṛ *tapara* which means that the *ṛ*

represents only those sounds of its class that are of the same time, in accordance with the sūtra *taparas tatkālasya* (1.1.69), in exception to 1.1.68 which allows a vowel to refer to all sounds of its class. If the *Ṇ* in the pratyāhāra *aṆ* in 1.1.68 were the first *Ṇ*, it would not have been necessary to mark *ṛ* as *ṛt* in 7.4.7. The very presence of the sūtra 7.4.7 therefore indicates that *ṛ* is a member of the *aṆ* in 1.1.68, and hence the *Ṇ* in 1.1.68 is the 2nd *Ṇ*.

5.2.4 *Lāghava* (economy)

Finally in case of *iṆ*, it is observed that when Pāṇini wanted to mention the 1st *Ṇ*, only two phonemes *i* and *u* being involved, he used *yvoḥ* instead of *iṆaḥ*. In fact, *yvoḥ* = *y v o ḥ* involves 3.5 (=0.5 + 0.5 + 2 + 0.5) morae (*mātrā*s, the time measure of utterance of a phoneme) whereas *iṇaḥ* = *i Ṇ a ḥ* involves 3 (= 1 + 0.5 + 1 + 0.5) mātrās. Thus in spite of prolixity (*gaurava*) of 0.5 mātrā, Pāṇini prefers *yvoḥ* over *iṇaḥ*, so that one always understands 2nd *Ṇ* in all other cases, achieving *lāghava* (economy) in other cases.

5.2.5 Why repetition?

Patañjali at the end of the discussion on this topic in the *Mahābhāṣya* raises a valid question: was there a dearth of consonants that Pāṇini used the same phoneme twice? In response he warns

vyākhyānato viśeṣapratipattiḥ na hi sandehād alakṣaṇam

(If one has got a doubt, one should not jump to the conclusion that the sūtra is defective. One should seek additional information from the commentaries.)

At the surface level, by repeating Ṇ, no doubt an ambiguity is introduced. Pāṇini's Aṣṭādhyāyī as several other Sanskrit texts do, does not carry any introduction or preface to his work explaining the purpose of his work, the methodology he used, etc. In the absence of any explanation by Pāṇini, on the repetition of Ṇ, we are forced

to conclude that **Pāṇini must be fully aware of the ambiguities a natural language has and also different sources of information such as sāmarthya, liṅga, lāghava, etc. available for disambiguation, and therefore in this particular case, might have allowed the repetition of the consonant.**

5.3 Pāṇini's subtle observations regarding Information coding in Sanskrit

Though a substantial part of the *Aṣṭādhyāyī* deals with the rules related to morphology, phonology and sandhi, an important section of it deals with concepts important from the language analysis point of view. Two of the important sections are those related to *kāraka* and *samāsa*. It is the *kāraka - vibhakti* mapping that provides a bridge between semantics and syntax. Both the kāraka and samāsa sections of Aṣṭādhyāyī focus on the information content and the coding a language adapts. We show here with examples from the kāraka section, the importance Pāṇini has given to information coding in a language string, while developing the theory of language analysis.

We produce three evidences from Pāṇini's *Aṣṭādhyāyī* where Pāṇini makes subtle observations about the information coding in a sentence.

5.3.1 Anabhihite

Consider the following three Sanskrit sentences:

Skt: *rāmaḥ grāmam gacchati.* (1)

gloss: Rama{nom} village{acc} go{active_voice,pr_tense,3_person,sg} (1a)

Skt: *rāmeṇa grāmaḥ gamyate.* (2)

gloss: Rama{instr} village{nom} go{passive_voice,pr_tense,3_person,sg} (2a)

and

Skt: *gacchāmi. / gacchasi.* (3)

gloss: go{active_voice,pr_tense,1/2_person,sg} (3a)

A typical computational linguist would say,

- In case of an active voice, a kartṛ gets a nominative case and a karma gets an accusative case.
- In case of a passive voice, kartṛ gets an instrumental case and karma (in case of transitive) gets a nominative case.
- kartṛ(karma) and the verbal suffix agree in number and person in active(passive) voice.
- Sanskrit also allows first person and second person pronoun drop.

This is fairly a good attempt to describe various phenomena observed in the above sentences. However, just as for a vaiyākaraṇa brevity is important[4], for a computational linguist not only the solution but also its optimality and generality matter the most. Optimality ensures that it consumes optimal time and space, whereas generality ensures that the same code will work for other languages as well. Therefore for a computer scientist, who is looking at the dynamics of information coding in a natural language, it becomes important to know 'where' exactly is the information about the kāraka roles coded?

Pāṇini handled the four cases described above in a very compact and elegant way. He gave the following sūtras:

[4]ardhamātrā lāghavena putrotsvamanyante vaiyākaraṇāḥ

1. *laḥ karmaṇi ca bhāve ca akarmakebhyāḥ (kartari) 3.4.69*
2. *anabhihite 3.1.1*
3. *kartṛkaraṇayoḥ tṛtīyā 2.3.18*
4. *karmaṇī dvitīyā 2.3.2*
5. *prātipadikārthaliṅgaparimāṇavacanamātre prathamā 2.3.46*

3.4.69 says that it is the *lakāra* (tense-aspect-modality marker) which expresses the *kartā*, *karma* or *bhāva* (action).

Having said this, now Pāṇini starts a section on mapping the *kāraka* relations into *vibhakti*s with the sūtra *anabhihite*[5](if not already expressed). In case the relation has not been expressed by any of other means, then the rules from 2.3.2 to 2.3.73 come into effect and the unexpressed *kāraka* relations are expressed through the *vibhakti*s. Then naturally, one would ask what does then the nominative case signify? According to Pāṇini (2.3.46) the nominative case just indicates the gender, number etc. and not any *kāraka* relation.

We see that Pāṇini deviates from the natural expectation in two ways.

- He does not give two different rules for active and passive. But handles both by a single rule. [55].
- This he achieves, by his minute observation: which information is redundant and which is not. It is natural to think of vibhaktis associated with nouns as

[5] *Kātyāyana* in his *vārtika* on this sūtra states that there are 4 ways by which the *kāraka* relations can be expressed – by means of *tiṅ* suffix, *kṛt* suffix, *taddhita* suffix (derivational suffix deriving a noun) and *samāsa* (compound).

marking the relations. But with his 'intuition for laguage analysis', he categorically denies any 'information content related to the marking of relations' in the suffix denoting 'prathamā vibhkati', and claims the presence of relation marking information in the verbal suffixes.

What we learn from the way Pāṇini framed the rules is to look for **where** the information is coded. The very fact that language allows absence of first and second person pronoun triggers that it is the verbal suffix which codes the *kāraka* relation and not the nominative case.

Many a times a language has redundant information. It is necessary to identify which part of it is redundant and which part of the coding is genuine. The question "where does a language code information?" thus helps us in ruling out the redundant information thereby helping one to build a NLP system that is more reliable and robust.

5.3.2 How much information is coded

In the previous section we saw that the *vibhakti*s (case markers) are determined by the *kāraka* relation a noun has with respect to the verb and the *prayoga*(voice). So we may express it as

$$vibhakti = \text{f } (kāraka, prayoga).$$

Vibhkati (case marker) and the *prayoga* (voice) are the surface level realities. *Kāraka*s are the basic syntactico-semantic categories. These categories, "serve as intermediaries between grammatical expressions and their semantics" (Cardona,1978) providing a bridge between the surface form and its meaning.

We argue below that Pāṇini, **by way of introducing an intermediary level of analysis draws a line between what is coded in a language string and what**

is extra-linguistic.

Look at the sentences

1. *rāmaḥ kuñcikayā tālam udghāṭayati.*

2. *kuñcikā tālam udghāṭayati.*

3. *tālaḥ udghāṭyate.*

Semantically speaking, in the above sentences, *rāma* is an agent, *kuñcikā* is an instrument and *tālaḥ* is a goal. However, according to Pāṇinian analysis all of them are kartṛ. It is obvious that by calling all these three *kartā*s, the actual semantic roles are not captured and one needs one more mapping from these *kāraka* roles to the thematic roles to arrive at the semantics. Natural question is then why Pāṇini did not go for the semantic analysis? And why did he chose the *kāraka* level analysis? Pāṇini observes

svatantraḥ kartā (1.4.54).

An activity may involve more than one participant. The underlying verb expresses the complex activity which consists of sub-activities of each of the participants involved. For example, in case of opening of a lock, three subactivities are very clearly involved (Bharati,1995), viz.

1. the insertion of a key by an agent,

2. pressing of the levers of the lock by an instrument (key), and

3. moving of the latch and opening of the lock.

Though in practice, to a large extent all three subactivities 1 through 3 together constitute the activity 'opening a lock', sometimes the subactivities 2 and 3 together are also referred to as 'opening a lock' and the activity 3 alone is also referred to as 'opening a lock'. Different languages may or may not have different lexical items

expressing these subactivities. When we say *rāma*, *kuñcikā* and *tālaḥ* are the *kartā* of opening of a lock, *rāma* is the *kartā* of the complex activity 1 through 3, *kuñcikā* that of 2 through 3 and *tālaḥ* that of 3 alone.

Patañjali, in Mahābhāṣya, interprets *svatantraḥ kartā* as: In a complex activity consisting of subactivities a_1 through a_m, if the speaker does not intend to mention participants capable of performing activities a_1 through a_j ($j < k$), the participant initiating the subactivity a_k will be the *kartā*.[6]

Thus in the absence of an agent (*rāma*), by promoting an instrument (*kuñcikā*) to *kartā*, Pāṇini draws our attention to the fact that language does not code information completely. Information related to the semantic encoding is not coded in a language string. To arrive at the conclusion that *kuñcikā* is an instrument and *tālaḥ* is a goal, one has to appeal to the world knowledge. The greatness of Pāṇini lies in **"identifying exactly how much information is coded and then giving it a semantic interpretation"** (1.4.23 - 1.4.55). This level of semantics is the one which is achievable / reachable through the grammar rules and the language string alone. This puts an upper bound for the analysis, making it very clear what is guaranteed and what is not. We can extract only that which is available in a language string 'without any requirement of additional knowledge'. To give an analogy, one can not use low quality energy to do the high quality work.

5.3.3 How (manner) is the information coded?

The *sup* and *tiṅ* suffixes assign *kāraka* roles to the nouns. The principles governing the relations between these suffixes with the *kāraka* roles are as under [56].

[6]Patañjali on kārake 1.4.23: evaṃ tarhi pradhānena samavāye sthālī paratantrā, vyavāye svatantrā|tadyathā amātyādīnām rājñā saha samavāye pāratantrya.m vyvāye svātantryaṃ ||(in the absence of a king, the senior most minister will enjoy the powers of king.)

1. Every *kāraka* must be expressed by a morphological element.

2. No *kāraka* can be expressed by more than one morphological element.

3. Every morphological element must express something.

Now consider a sentence

Skt: *rāmaḥ dugdham pītvā śālām gacchati.* (4)
gloss: Rama{nom} milk{acc} after_drink{gerund} school{acc} go{pr,active,3p,sg} (4a)
Eng: Rama went to school after drinking milk.

In this sentence, there are two verbs viz. *gam* and *pā*. Both of them have a mandatory expectancy of two *kāraka*s viz. *kartṛ* and *karma*. Further the relation between the subordinate verb and the main verb should also be marked. Thus there are 5 relations which need to be marked. In the above sentence, there are 5 words and hence only 4 relations can be expressed through the suffixes. Relations that are expressed by the suffixes are shown in Figure 5.1.

The *kartā* of the verb *pā* is not marked explicitly. A native speaker, however, does not have any problem in answering the question 'who drank the milk?'. This indicates that it is the 'Language Convention' that tells: in case of *ktvā* suffix[7] the *kartā* of the subordinate verb is the same as that of the main verb. Pāṇini has postulated this in terms of a sūtra

samānakartṛkayoḥ pūrvakāle (3.4.21)

It is the **language convention which gives a license to not to code the information explicitly**. The implicit coding of the information may need extra processing

[7]which indicates that the action corresponding to the verb with *ktvā* suffix takes place before the action indicated by the main verb.

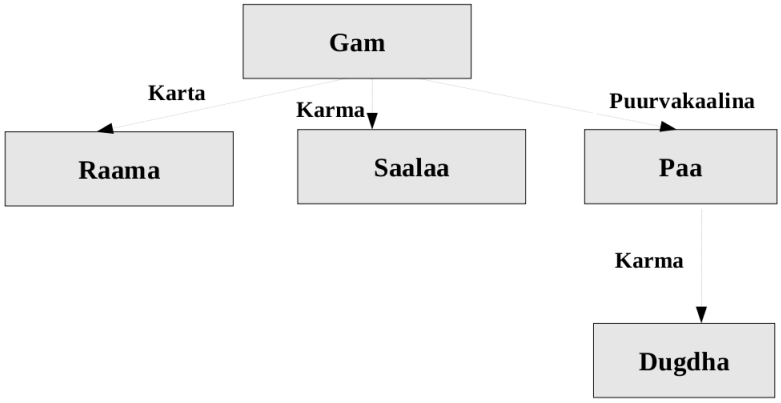


Figure 5.1: modifier-modified relations

for making such a knowledge explicit. It then becomes crucial for MT developers to know what is coded explicitly and what is coded implicitly. If the two languages have different language conventions, one needs to make implicit information explicit in other language. This may lead to unacceptable constructions, or even to a catastrophe, if not handled properly.

Consider

Skt:*vanāt grāmam adya upetya odanam āśvapatena apāci* [56]. (5)

gloss: forest{abl} village{acc} today after_reaching rice{nom} Asvapata{inst} cook{passive, past, 3pr,sg}. (5a)

Eng: Today after reaching from a village, the rice was cooked by Asvapata.

In Sanskrit, following the sūtra P 3.4.21 viz. *samānakartṛkayoḥ pūrvakāle*, it is clear that it is *āśvapata* who returned and it is he who cooked. But such constructions are not allowed in English. English needs passive absolutive, if the finite verb is in passive. Sanskrit uses the same '*ktvā*' in both the active as well as the passive constructions. Hence in MT they pose a problem, as they may lead to unacceptable / ungrammatical constructions.

5.4 Conclusion

With the emergence of modern Linguistics, linguists had started recognising the importance of Pāṇini's grammar. And now with the advent of computer technology, computational linguists have started recognising Pāṇini as an information scientist.

The information coding and flow of information are at the center of the Pāṇinian analysis. The questions where does a language codes information, how much information does it code, and the manner in which it codes the information are the three aspects of the information dynamics or the parameters that are crucial in identifying the "true nature of the language". These three parameters may be used to determine the syntactic divergence between the languages. And hence we claim that any grammar which is developed with the three questions in mind: **where**, **how much** and **how** is the information coded, would be truly in Pāṇinian spirit. In the next chapter we look at the English language, from the point of view of information coding. In this process, naturally we compare English structures with those of Hindi which leads to the study of structural divergences between English and Hindi.

Chapter 6

English from Hindi viewpoint: A Pāṇinian Perspective

6.1 Introduction

With the world wide web spreading all over the world, information is now available at the click of a mouse. Most of this information is in English. In India hardly 10%[1] of the population can understand English. Thus the *language barrier* leads to a *digital divide*. Hence, if India has to take real advantage of the new technology, it is necessary to make this information available to the Indians in their native languages. Experiments have shown that English-Hindi anusaaraka, as a tool to access English text has been useful [13]. Since the top layers of the anusaaraka output, produce an output that follows the grammar of the source language, it is necessary for a serious user to 'learn' the 'contrastive grammar' between the source and the target language. A courseware, explaining the differences between English and Hindi will therefore be needed to access anusaaraka.

[1]source http://en.wikipedia.org/wiki/List_of_countries_by_English_speaking_population

In Chapter 5 we have seen the information centric point of view of Pāṇini towards analysis of a language. In this chapter we use Pāṇinian way of analysis to discover the structural differences between English and Hindi. This study will not only be useful for an anusaaraka reader, but also to the Machine Translation community working on English-Hindi translation systems, since it will throw open the problematic cases in English to Hindi Machine Translation.

6.1.1 Traditional view

The structural differences between English and Hindi are mostly attributed to the differences in their word orders. Language typologists[8] classify English as an SVO language and Hindi as an SOV language. However comparing English and Hindi on the basis of word order is like comparing apples with oranges! The reason is: English uses position to code crucial information of the relation between the words in a sentence. So when one says English is an SVO language, one is asserting a fact about the encoding of grammatical relations, viz. subject and object, with respect to verb in English. The position immediately preceding a verb marks the subject and the one immediately following marks an object. On the other hand, in case of Hindi, a relatively free word order language and claimed as an SOV language, one is just stating a statistical fact about the order of words in a typical Hindi sentence!

To make the point clear, the following two English sentences have exactly opposite meanings.

Rats kill cats. (1)

Cats kill rats. (2)

whereas, the following two Hindi sentences with similar change in the order of words as above, have the same meaning (ignoring the topicalisation, of course).

Hnd: rāma phala khātā hai. (3)

gloss: Ram fruit/fruits eats. (3a)

Eng: Ram eats a fruit/fruits. (3b)

Hnd: phala rāma khātā hai. (4)

gloss: Fruit/fruits Ram eats. (4a)

Eng: Ram eats a fruit/fruits. (4b)

In the following sections, we investigate the reasons behind the structural differences between these two languages from information coding point of view. In the second section, on the basis of the basic structure of declarative sentences in English and Hindi, we conclude that English does not have a morphological formative for an accusative marking. The missing accusative marker is compensated by freezing the subject position in English. As a consequence, this fixed subject position gives rise to some structural differences between the two languages. The third section discusses these structural differences. In the fourth section, it has been pointed out that, English does not have a morphological formative marking the yes-no question. Therefore English resorts to the word order again. The structural differences arising because of this phenomenon are discussed in the fifth section.

6.2 Missing accusative marker

Look at the following English sentence and its Hindi gloss

Eng: Rats kill dogs. (5)

Hin gloss: cūhe māra{0}[2] kutte. (5a)

[2]{0} stands for no overt suffix. The verbal form with 0 suffix in English has more than one interpretations – non 3^{rd} person singular, or to-less infinitive, etc. The one which is relevant in this particular case is present tense non 3^{rd} person singular.

One may interpret the above Hindi gloss as

cūhe mārate_haiṁ kutte. (6)

Though this is not a grammatical Hindi sentence, still, if a Hindi reader is asked to interpret this sentence, he will interpret this as

Hnd: kutte mārate_haiṁ cūhoṁ_ko. (7)
gloss: dogs kill rats{acc.} (7a)
Eng: Dogs kill rats. (7b)

which is exactly the reverse of what is being said in English!

Why does this happen? First let us try to understand the reason for why a Hindi reader analyses it in this way, and later we will see what mechanism in English triggers the desired meaning.

6.2.1 Arguments by Hindi reader for this interpretation

The Hindi sentence (5a) is ungrammatical, because Hindi requires an accusative marker (*ko*) to mark the *karma* role. However, Hindi also has a tendency to drop the *karma vibhakti*, wherever there is a possibility of recovering the information from other sources, such as world knowledge. For example, consider the sentence (8) below.

Hnd: rāma phala khātā hai. (8)
gloss: Ram fruit eats. (8a)
Eng: Ram eats fruits. (8b)

In sentence (8), *phala* does not have an accusative marker. In spite of this, a Hindi

reader appealing to the world knowledge (in this case the *yogyatā* or competency), interprets this sentence as *rāma* is the *kartā* of the action of eating and *phala* is the *karma* of the action.

At the same time, Hindi obligatorily requires an accusative marker, if anything against *yogyatā* is to be communicated, as is obvious from the following Hindi sentence.

Hnd: latā śarāba nahīṁ pītī, śarāba latā ko[3] pītī hai. (9)

Gloss: Lata wine not drink, wine Lata_{acc.} drinks. (9a)

Eng: Lata does not drink (consume) liquor, liquor drinks (consumes) Lata.

Following the same argument, since sentence (5a) does not carry any explicit accusative marker to mark the *karma*, Hindi reader appeals to his world knowledge, since the dogs have *yogyatā* to kill the rats and not the other way, interprets the sentence as (6).

Unlike Hindi, English does not have an explicit morphological formative for accusative marking. Rather it codes the information to indicate grammatical relations of *subject* and *object* by their positions.

Now one may ask the question, which position is crucial, the subject position or the object position or both? In other words, what is invariant, the S-V order or the O-V order or the S-V-O order?

6.2.2 Initial Hypothesis (S-V-O order)

`Both the subject as well as the object positions carry crucial information, hence the S-V-O relation is invariant in English.`

However, we come across such sentences as

[3]the marker *ko* with latā, is to mark latā as a *karma*. This *ko* can not be dropped.

- Who likes sweets? **Sweets**, I like. (10a)
- Mrs. Venables turned a little pale.

  Lord Peter presented no difficulties,

  but **Bunter** she found rather alarming. (10b)

  (source: D. Sayers, The Nine Tailors)

In sentences (10a) and (10b), the object is in the topic position, i.e., topicalised to mark the contrast.

Therefore, in this case, the information that **sweets** is an object of the verb **like**, is not coded in position or the order. In other words, the orders S-V-O as well as V-O are not invariant.

6.2.3 Revised Hypothesis (S-V order)

This leads us to reframe the observation as,
`It is the subject-verb order which is invariant. Objects may move around.`

But life is not simple as is expected. There are examples showing movement of subject also!
Here are the examples:

Uneasy *lies* **the head** that wears a crown. (11a)
Never *was* **the sea** so calm! (11b)
Here *comes* **the bus**! (11c)
On the bed, *hung* **a mosquito net**. (11d)

In the above examples, **the head, the sea, the bus** and **a mosquito net** are the subjects of the verbs **lies, was, comes** , and **hung**, and do not precede the verbs.

However, in all these examples, the verb is monovalent. That is, they have an expectancy of only one argument, which agrees with the verb in number and person. Therefore, its position in a sentence is not crucial. However, in case of transitive verbs, there are two arguments, and hence it is necessary to mark at least one of them. From the above examples, what we observe is:

`In case of transitive verbs, subject is always to the left of the verb, or in other words, S-V order is invariant!`

There is evidence, which goes against this hypothesis too. Look at the following sentences

Something had to give. And **give it** did. (12a)
Last October our good friend in South Africa, wanted to come to England this year and **come he** did, with his wife Annie. (12b)
We all said she was bound to leave him, and **leave him** she did. (12c)
She could only hope that Harriet was mistaken in his feelings...
Wish it she must, for his sake... (J. Austen, Emma) (12d)
Ride in a taxi with Pamela and Bredon **he** could not, even if it meant losing her forever... (D. Sayers, Murder must Advertise) (12e)

In all these examples, the subject is after the verb phrase! But at the same time, we also note that subject is followed by an auxiliary! Finally it precipitates (to the observation) that it is the subject-auxiliary proximity (*sannidhi*) that is invariant in English. The normal proximity between auxiliary verbs and the main verb gets violated in English, and a new proximity is established between a subject and an auxiliary verb.

This leads to a concept of 'Subject Position' - a position which is to the immediate left of the auxiliary verb, or the main verb (in case auxiliary verb is absent). And

thus, we revise our observation as:

6.2.4 Final Observation

`In case of transitive verbs, the missing accusative marker in English has been compensated by a position called 'Subject Position'.`

We see below the consequences of the missing accusative marker.

6.2.5 Consequences of missing accusative marker

Difference in word order

Major consequence of coding the information in terms of position is reflected in the difference in word order.

- Hindi has post positions, while English has pre-positions

  e.g. **at** the door -> daravāje_**para**

  Leaving apart the subject position, the remaining part of English sentence structure reflects the mirror image of the corresponding Hindi sentence structure.

  e.g. look at the English sentence and its Hindi translation in table 6.1.

Eng:	There$_1$	is$_2$	somebody$_3$	knocking$_4$	at$_5$	the$_6$	door$_7$
Gloss:	0	hai	koī	khaṭakhaṭātā huā	para	–	daravāje
Hnd:	daravāje$_7$	para$_5$		khaṭakhaṭātā_huā$_4$	koī$_3$	hai$_2$	.

Table 6.1: English-Hindi: Mirror reflection

- English uses phrasal verbs formed of verbs followed by particles giving different shades of meaning. These particles are in post verb position, whereas the upasargas in Sanskrit (and in Hindi also) have the same role, but are used as prefixes. For example,

  Eng: look at, look for, look after, etc.

  Hnd: āhāra, vihāra, prahāra, etc.

- The order between main verb and the auxiliary is reversed. In Hindi the verb groups are formed by the main root followed by auxiliary verbs, as in *jā rahā hai*. However, in English the order is: auxiliary verbs followed by the main verb as in *is going*.
- It should be noted that the reversal is not everywhere. For example, in case of word formation, at morphological level, the order is still root or stem followed by a suffix, as we see in the words *going, goes, chairs*, etc. Similarly in case of noun groups, the relative order of adjective and noun is same as in Hindi. For example: compare *the red book* with *lāla pustaka*.

Exceptional Case Marking (Subject-Object raising)

Another phenomenon in English is raising of subject to object position or also termed as exceptional case marking (ECM).
Consider the sentence:

Eng: I want [him to go there]. (13)
gloss: maiṁ cāhatā hūṁ usako jānā vahāṁ. (13a)
From Hindi point of view, there are two problems in this sentence.

- In case of icchārthaka (indicating desire etc.) verbs in Sanskrit (and also in most Indian languages), if the subordinate verb has *tumun* (infinitive) suffix, then it shares the kartā with the icchārthaka dhātus (*samānakartṛkeṣu tumun*, Pāṇini: 3.3.158). For example,

  Skt: aham bhoktum icchāmi. (14)
  gloss: I eat_to desire{1p sg}. (14a)
  Eng: I want to eat. (14b)

Hence, to express a reading such as

I want [him to go there]. (15)

wherein the *kartā* of want is different from the *kartā* of *go*, Hindi can not use nā (tumun) construction to convey this reading.
Hindi either uses a finite clause such as

maiṁ cāhatā_hūṁ ki vaha jāye. (16)

or, uses non-finite clause such as

maiṁ usakā jānā cāhatā hūṁ. (17)

- Second problem with this construction is, this structure is inherently ambiguous. Consider the sentence,

  I want this pen to write. (18)

  The most likely meaning is

  I want [this pen] [to write]. (19)

  and not

  I want [this pen to write]. (20)

  as in (9). It is the world knowledge which triggers the desired meaning.

The question is what phenomenon in English does make such constructions inherently ambiguous? Secondly why the *he* which is the subject of *go* has an accusative marker?

Answer to the first question is not very difficult. It is the subject of the infinitive verb which also happens to be the object of the main verb that makes the

sentence inherently ambiguous.

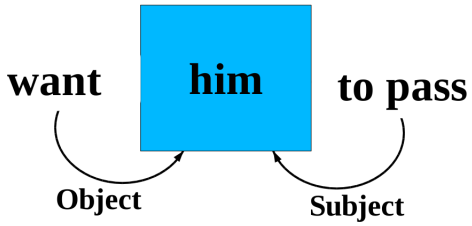


Figure 6.1: ECM phenomenon in English

This construction in generative grammar has been termed as Exceptional Case Marking (ECM) or raising to object phenomenon. *He*, which is at the subject position of *go*, gets case marked by the verb *want*. This is because, *go* not being in finite form, can not assign case to its arguments whereas, *he* being also in the object position of *want*, *want* can assign a case to *he*, making it *him*. Here it should be noted that the word *him* should not be analysed as *he* + accusative marker. *Him* here is just an oblique case of *he*, a morphological requirement. Thus, a verb is assigning case to something which is not its own argument. That is why such verbs are referred to as Exceptionally Case Marking verbs. This phenomenon is also sometimes referred to as subject-object raising, since the subject of the subordinate clause has been raised to the object position of the main verb. This explanation makes one feel that the sentence formation is *asamartha*, and in spite of it, a native speaker finds the sentence to be good. So there should be some *gamakatva* (the ability to convey the desired meaning) because of which an English reader is able to get the correct meaning. This *gamakatva* is in the subject position of the verb go.

Subject sharing (gapping)

The quest 'where' is the information coded in English lead us to the concept of *Subject Position*. 'How' is information coded is equally important. Whether the information is coded explicitly or implicitly has direct bearing on the parsers. If the information is coded explicitly, then it is available directly for parsing. However, if it is coded implicitly, following language convention, then one has to extract this information indirectly. Further, if other language does not follow the same convention, it may lead to catastrophe as well.

Here is an example where English and Hindi exhibit different behaviour in the sharing of kārakas. This may lead to improper or misunderstanding of English sentence by Hindi native speakers.

For example, consider the sentence
Eng: Mohan dropped the melon and burst. (21)
The Hindi gloss of this is
Gloss: mohana ne girāyā tarabūja aura phuṭā. (21a)

The *kartā* of *phutanā* is missing, and hence appealing to the *yogyatā*, a Hindi reader interprets this as

Hnd: mohana ne tarabūja girāyā aura tarabūja phuṭā[4]. (22)

whereas the English sentence (21) means
Mohan dropped the melon and **Mohan** burst. (23)

[4]Hindi, unlike English, does not have a subject sharing rule. Hindi allows such usages as rāma ne subaha kapade dhoye aura dopaharataka sukha bhī gaye.
In this sentence, the *kartā* in the second sentence (*kapade*) is same as the *karma* in the previous sentence.

Though this sentence may sound senseless to an English reader, still s/he can't get any meaning from this sentence other than the above one. To get the other meaning viz. that the melon burst, English has to use another construction viz.

Mohan dropped the melon and **it** burst. (24)

6.3 Missing yes-no interrogative marker

Look at the following two sentences in English:

Eng: Ram is going to school. (25a)
and
Eng: Is Ram going to school? (26a)

The first one is declarative and the second one is a yes-no question. If we look at the words in the sentences, they are the same, except for the word order. So it is natural that the information of 'interrogativeness' or 'declarativeness' of the sentence is in the word order. There is no explicit morpheme to mark the interrogativeness.

From the Hindi translations of these sentences

Hnd: rāma skūla jā rahā hai. (25b)
and
Hnd: kyā rāma skūla jā rahā hai? (26b)

it is clear that Hindi has an explicit word 'kyā' to mark the 'yes-no' question. The counterpart of this morpheme is missing in English. As a consequence, English codes this information in the form of ordering of words.

6.3.1 Observation

`The missing marker corresponding to yes-no question is compensated by the 'subject-auxiliary verb inversion' in English.`

As a consequence the normal proximity between auxiliary verbs and the main verb is weakened, and a new proximity is established between the subject and the auxiliary verb.

6.3.2 Consequences of Missing yes-no interrogative marker

- Subject Position can't be empty:

  For, if it were empty, it would not be clear whether the given sentence is interrogative or declarative.

- Insertion of auxiliary *do* in interrogatives: If a verb form does not involve an auxiliary verb, then a dummy *do* is inserted, as shown below.

  He goes to school. (27)
  Does he go to school? (28)

  Here *goes* is split as *does+go* by introducing an auxiliary *do*, and the auxiliary *does* then is inverted with the subject to give an interrogative sentence (28).

- extra overheads:

Since English codes information in Subject Position as well as in subject-auxiliary order, Subject position can't be empty. This forces English to bear an extra overload of dummy *it* and existential *there* to fill the Subject Positions.

– Dummy/expletive it

Consider the following English sentences

Eng: It is raining. (29)

Eng: It is very hot outside. (30)

and their Hindi translations

Hnd: bārīsha ho rahī hai. (31)

Hnd: bāhara bahuta garamī hai. (32)

Hindi translations do not have any counterpart of the dummy *it*. This *it* is termed as expletive or dummy *it*. As the name implies this *it* does not carry any information, and is just a place holder or a stand-by.

– Expletive There

Consider the following sentences

Eng: There are flowers in the garden. (33)

Eng: There could have occurred several riots. (34)

Eng: There are thought likely to be awarded several prizes. (English Syntax, Radford, pp 246) (35)

In these sentences, the subject position is occupied by the word *there* and the word with which verb shows agreement (*ukta*) is moved away from the subject position. This *there* is not an adverbial there, since one can say

Eng: There are flowers there. (36)

The second *there* in the above sentence is an adverbial *there*.

Hindi translations of above sentences are

Hnd: bagīce_meṁ phūla haiṁ. (37a)

Hnd: kaī daṁge ho_sakate_the. (37b)

Hnd: kaī pāritoshikoṃ kā vitaraṇa hone kī saṁbhāvanā vyakta kī jā rahī hai. (37c)

In Hindi translations we do not see any counterpart of English *there*.

There in all these cases is also called an existential *there*, since it expresses the existence or appearance.

Why does English require this existential *there*?

The words or phrases that are to be focused are normally moved into a focus position at the front of a clause in order to highlight it. When a verb is to be focused, it is not possible to bring it to the front, since it either renders the subject position empty or it may read as an interrogative sentence. Hence in such cases, the subject position is filled with an expletive *there*.

It is interesting to note that there are certain transitive verbs which use expletive *there*. For example,

Eng: There entered a hall an ugly old man. (Levin, 1993:p90) (38)

This *there* thus serves as a focus element to express the 'factuality' or 'happeningness' of the event.

6.3.3 Subject-Subject raising

Since the dummy *it* does not carry any lexical information, it is an extra overhead. So there is a tendency to drop it whenever such an opportunity exists. This is natural and consistent with the principle of economy. For example, consider the following English sentences.

Eng: It seems that the boys have eaten fruits. (39)

Eng: The boys seem to have eaten fruits. (40)

These two sentences are equivalent. The phenomenon where the subject of the subordinate clause has been moved to the subject position of the matrix clause, is known as subject-subject raising in western grammar,

This phenomenon is purely structural/syntactical and has nothing to do with the semantics. Once the subject of the subordinate clause is moved to the subject position of the main verb, the main verb shows agreement with this noun. Thus we see that, in sentence (40), the *boys* is in the subject position of *seem* and also is *ukta*, as it shows agreement with the verb *seem*. However, semantically, *the boys* is not a *kāraka* for the verb *seem*. It is the *kartā* of *eat*!

This inconsistency in the sentence structure thus should render the sentence incompetent (*asamartha*) to convey the desired meaning. Or in other words, the sentence has certain syntactic operations that do not convey any semantic connections. But

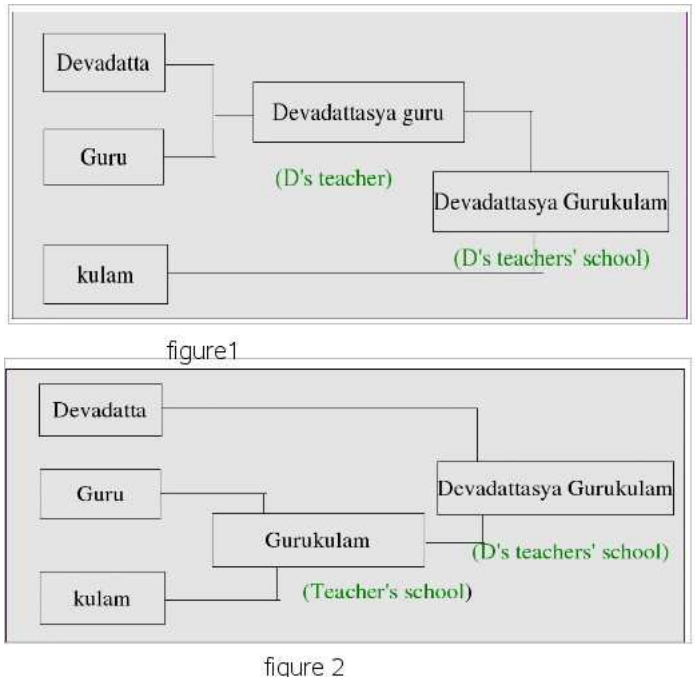


Figure 6.2: Sanskrit compound: Devadattasya Gurukulam

then, how is that an English reader does not find such constructions odd? or how is that the sentence is acceptable to the native speaker? How does a language allow proper communication in spite of apparent inconsistency at structural level to convey the desired meaning?

It may be said that it is the *gamakatva* (arthabodhakatva, the ability to convey the desired meaning) that takes care of proper communication. Patañjali has discussed this aspect under the commentary of *samartha padavidhiḥ* (Pāṇini: 2.1.1). He takes an example of Sanskrit compound *devadattasya gurukulam* (school of D's teacher). Devadatta is related to guru. So semantically, it should be analysed as in figure1 of 6.2. However, its syntactic decomposition is as described in figure 2 of 6.2.

This is possible only because, the word *guru* is sākāṁksha (having expectancy). Hence

even if it joins with other word to form a compound, this *sākāṁkshatā* (the property of having expectancy) makes it possible to convey the desired meaning. This *sākāṁkshatā* is the *gamakatva* which makes it possible to interpret the given compound properly.

On similar lines, in case of (40) the verb eat requires a subject. But its subject position is empty. At the same time, *seem* does not need any subject, and its subject position is occupied by *the boys*. The *gamakatva* is in the fact that *seem* does not require any subject. It expects only a clause following it.

So English native speaker seems to push the occupant of the subject position of the verb *seem* to the right towards the first subordinate verb whose subject position is empty. Thus for a native English speaker, the subject position is more important than the agreement between the verb and the subject.

6.3.4 Tough movement

There is a class of adjectives (tough, difficult, easy, hard, etc) which also exhibit a phenomenon of raising. This phenomenon is often referred to as tough movement. Look at the following pair of sentences.

Eng: It is hard to see John. (41)

Eng: John is hard to see. (42)

As is clear from the above example, the object (John) of the subordinate verb is moved to the subject position of the main verb (is) which was occupied by dummy *it*. This is again a case of asāmarthya (incompetency), since there is an agreement between the occupant of subject position (John) with the verb (is), whereas the oc-

cupant of subject position is not an argument of the verb!

Not only the objects, but even the complements of prepositions can also move to the subject position. However, when the complement of prepositions move to the subject position, the prepositions are left behind. This then gives rise to 'violation of normal sannidhi/expected proximity'. For example

Eng: This violin is tough to play these sonatas on. (43)
where, the normal sannidhi between preposition (on) and the noun (violin) is violated.

Further with the *believe* type of verbs, English allows constructions such as
Eng: John is tough to believe that University would fire. (44)
Eng: Students are tough to believe that University would fire. (45)

Here also, the agreement is just a structural requirement and does not carry any semantic information. The gamakatva is in the ākānkshā of the adjective *tough* and the verb in the subordinate clause.

6.3.5 Wh questions

The phenomenon of subject-auxiliary inversion is also observed in wh-questions, where the wh-element is brought forward to the topic position in order to focus. Hence, with an exception of subject of wh-questions (where the wh is already in focus), the wh-questions also show a 'subject auxiliary inversion'. Here are some examples:
Eng: Whom did Ram kill? (46)
Eng: Where did Ram go? (47)
with an exception of wh-questions on subject, as in
Eng: Who killed Ravana? (48)

Further in case of wh questions on NPs which are objects of a preposition, the prepositions normally are not moved along with the wh elements, as in

Eng: Who did Ram talk to? (49)

English does allow pied piping wherein the sannidhi between prepositions and it's object is intact as in

Eng: To whom did Ram talk? (50)

However, this order weakens the focus on wh element. The preferred order in English, is as in (49) with stranded preposition and not as in (50).

6.3.6 Inversion in tagged questions

English does not have any separate morpheme for tagged questions, hence again, it resorts to the subject-auxiliary inversion, as in

Eng: He has gone to Mumbai. Hasn't he? (51)

Eng: He won't win. Will he? (52)

6.3.7 Inversion in other constructions

Phenomenon of subject auxiliary inversion is seen in other constructs also.

Eng: Ram has gone to the market. So has Shyam. (53)

Eng: Only then, did he understand the joke. (54)

Eng: No other colleague, would I trust. (55)

Eng: Suffice it to say that ... (56)

(53) is an example of gapping phenomenon. It is also interesting to note how the

word order helps in triggering the correct sense of the word *so* between its two senses, viz. *therefore*, and *also*. Compare (53) with

Eng: Ram has gone to the market, so Shyam was waiting for him. (57)

(54) and (55) are examples of focus on the factuality, and (56) is an idiomatic expression.

6.4 Conclusion

The purpose of the foregoing exercise is to look at the structural differences between English and Hindi from an information theoretic point of view. The major reason behind the structural differences between English and Hindi is the absence of accusative marker and yes-no marker in English. To compensate for this absence, English resorts to the word order. This further gives rise to more structural differences between the two languages. The interaction between different phenomena have been explained in figure 6.3.
We conclude that a Hindi reader while reading an English text has to 'tune' himself to the following:

- Acquire a new 'vṛtti' – the 'quazi compound' _V_,
- Do away with the normal 'sannidhi' (proximity) between a verb and its auxiliary and also between a noun and its post-position (which are integral part of Indian languages), and acquire new 'sannidhi's between:

  1. a subject and auxiliary and
  2. a verb and its preposition,

  and finally,

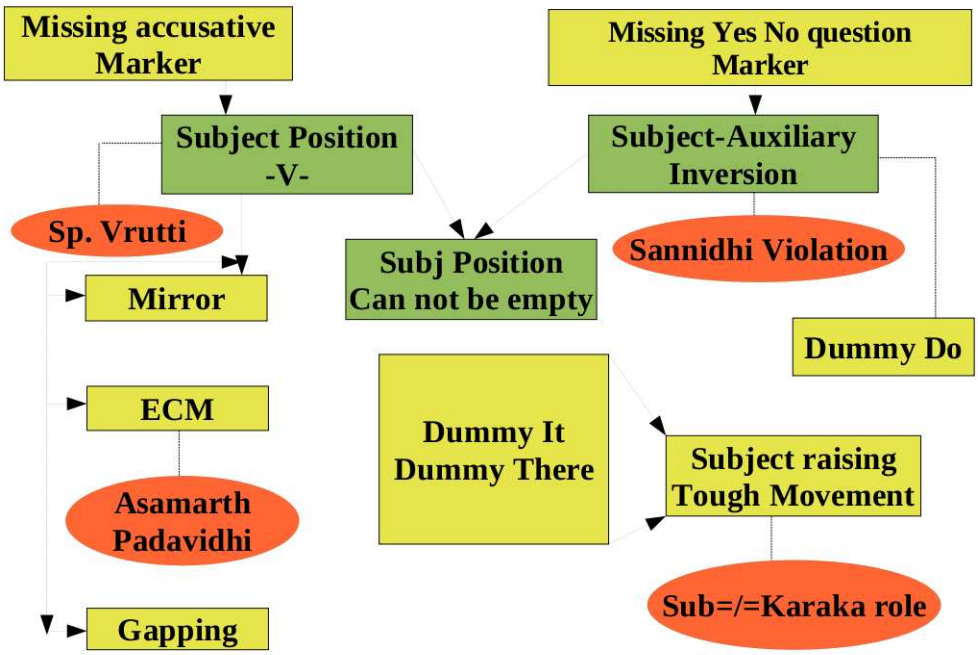


Figure 6.3: Contrast between English and Hindi

- The occupant of subject position need not have any kāraka role with the corresponding verb.

We discuss, with examples, how this grammar helps an anusaaraka reader to understand the text, in chapter 8.

Chapter 7

Dorr's Divergence and Anusaaraka

7.1 Dorr's Divergence

In Chapter 3 we have discussed the divergences between English and Hindi from the information coding point of view. The much discussed translation divergence is due to Dorr [32]. Dave et al [30] discussed these divergences with respect to English and Hindi, and Goyal et al [37] discussed them with respect to English, Hindi and Sanskrit. In this chapter we will look at these divergences to see how anusaaraka handles these divergences.

Dorr classifies the divergences into two major classes viz. syntactic and lexical semantic. These are further classified as:

- Syntactic divergence
  - Constituent Order divergence,
  - Adjunction Divergence,
  - Preposition stranding Divergence,
  - Movement Divergence,

  - Null Subject Divergence,
  - Dative Divergence and
  - Pleonastic Divergence.

- Lexical Semantic Divergence:
  The lexical semantic divergence [32] discussed in Dorr and Dave et al [30] in particular with reference to English and Hindi are
  - Conflational Divergence,
  - Structural Divergence,
  - Categorial Divergence,
  - Head swapping Divergence and
  - Lexical Divergence.

7.2 Anusaaraka solution

Anusaaraka being a language accessor, we look at these divergences from comprehensibility point of view. The important layer of anusaaraka is the first layer i.e. the layer of śabdasūtra. We address the issue of divergence with respect to this layer. In case, this layer can not handle the divergence leading to uncomprehensible output, we show how the next layer(s) handle it.

- Syntactic divergence
  - Constituent Order divergence:
    This divergence conerns with the word order in English and Hindi. As we saw earlier in chapter 6, we train the anusaaraka reader in the special vṛtti _V_ English exhibits. So in case of parser failures, the reader can still resort to the śabdasūtra layer and understand the meaning.

– Adjunction Divergence:

This is associated with the positioning of the adjective phrase. Hindi is flexible with two types of adjective phrase constructions, viz.

Hnd: vaha ladakā jo kala yahāṁ āyā thā (1)

gloss: the boy who yesterday here came

Hnd: kala yahāṁ āyā huā ladakā (2)

gloss: yesterday here came boy

whereas English has only one construction viz.

Eng: The boy who came here yesterday (3)

which corresponds to (1). Thus English does not have an equivalent construction for (2).

So when we translate from English into Hindi, there is no problem. Problem arises when we translate from Hindi into English. Hindi allows an adjectival participial construction, and English does not have a parallel construction. This case is exactly parallel to the case of missing adjectival participials in Hindi corresponding to the ones in Dravidian languages. Anusaaraka [70], discusses a solution to this problem by suggesting a special 'jo*' construction in Hindi.

– Preposition Stranding:

We saw in chapter 6 (examples 43 and 49) that there is more affinity between a preposition and a verb than between a noun and its preposition, leading to preposition stranding. This is thus a part of training on special sannidhi in English between verb and its argument markers.

– Null Subject Divergence:

Hindi is a null subject language whereas English obligatorily requires a subject. So translation from Hindi to English poses a problem. But through anusaaraka complete information is available. For example, look at the following anusaaraka output from Hindi into English.

Hnd: jāūṁgā. (4)

anu_eng: will_go{1p,masc,sg}.

Though the subject is missing, for a anusaaraka reader, the information regarding the subject is still available indirectly through the verb features, and as such this does not pose any problem, at the level of śabdasūtra.

– Pleonastic Divergence:

This divergence occurs with syntactic constituents having no semantic contents such as dummy *it*, *there*, etc. The anusaaraka reader is trained for this divergence giving logical reasons behind the presence of this divergence, or rather presence of dummy *it* and expletive *there*.

- Lexical Semantic divergence

  – Conflational Divergence

  This divergence arises when lexicon incorporates periphrastic mechanism to fill in gaps in lexicalisation describing an action. For example *stab* means *to strike with a knife*. Hindi does not have a word conveying the same meaning, and thus uses a phrase *chūre se māranā*, which is a literal translation of *to strike with a knife*. At śabdasūtra level, it is perfectly all right having a phrasal equivalent for a word. In fact the śabdasūtra may be much more complex than just a phrase.

  – Structural Divergence

  Consider a sentence

  Eng: He entered the house. (5)

  Hnd: usane ghara_meṁ praveśa kiyā. (5a)

  Here *the house* is a NP, whereas the phrase in Hindi *ghara_meṁ* is a prepositional phrase. This is taken care of in anusaaraka by vibhakti transformation rules for different verb frames. For example a transformation rule

in this case will be

enter → *praveśa karanā<obj:meṁ>*

– Categorial Divergence:

This arises because of change in the lexical category during translation. The example discussed in Dave [30] is

Eng: They are competing. (6)

In anusaaraka, *are competing* will be treated as a verb group, with *compete* as the verbal base and *are_ing* as a suffix. Thus the categorial divergence problem disappears!

– Head Swapping:

Under this, two cases of swapping viz. demotional and promotional swapping are discussed. The examples discussed in Dave [30] are:

Eng: It suffices. (demotional head swapping), and (7)

Eng: The play is on. (promotional head swapping). (8)

In both of these cases, anusaaraka does not see any problem. In *It suffices*, *it* is dummy. Thus, with the meaning of *suffice* as *kāphī honā*, there is no problem of head swapping. Similarly, in the sentence *The play is on*, the śabdasūtra layer may not give a comprehensible output, since this is a kind of idiomatic expression with preposition providing the aspect of continuity, treating *is on* as a meaningful unit, the word group level provides the desired output. Thus there is no question of Head swapping.

– Lexical Divergence:

The example discussed under this category, in Dave [30] is

Eng: John broke into the house. (9)

This also is not problematic. At the level of word grouping, the particle *into* will be grouped with the verb *break* to yield *break_into* as a meaningful unit, and providing its Hindi equivalent.

Thus we see that all these divergences can be taken care of at the śabdasūtra level or at the level of word grouping.

7.3 Problematic Cases

There are certain typical syntactic construction in English, which cause a divergence between English and Hindi. They are:

- Resultative constructions such as

  She washed the plates clean. (10)

  The river froze solid. (11)

  The gardener watered the flowers flat. (12)

  They painted the house green. (13)

- Verbs of motion specifying the manner of motion as in

  She danced into the room. (14)

  She smiled her thanks. (15)

  She beamed her adoration. (16)

- The absolutive constructions such as:

  Julie accepted the award, tears streaming down her face. (17)

  The cat confronted the burglar, its back arched and teeth bared. (18)

In all such cases, plan is to provide an online intelligent help. The interface will be intelligent enough to identify the difficult constructions and provide necessary help.

Chapter 8

Pravṛtti-nimitta and Śabdasūtra

8.1 Pravṛtti-nimitta

In Indian Grammatical Tradition, we see abstractions at phonological, morphological, and morphophonemic level as well as at semantic level. The abstraction of the Sanskrit words rāmeṇa and ramayā as rāma + ṭā and ramā + ṭā is an example of suffix level abstraction. Postulation of dhātus (verbal bases) in various classes gaṇas of the dhātu-pāṭha is an example of abstraction at the level of morphology. We find similar abstraction at the level of semantics in the concept of pravṛtti-nimitta.

All the Indian schools admit a relation between a word and its meaning. This relation is termed as śakti. However, these schools differ in the nature of this relation. According to the mīmānsakas, the relation is natural[1]. The Grammarians also accept this view[2]. They believe 'just as the organs of perception have an innate capability to perceive, similarly the words have an innate capability to express the meaning[3]'. The

[1]mīmānsāsūtra 1.1.5 : autpattikas tu śabdasyārthena sambandhaḥ.

[2]siddhe śabdārthasambanddhe – Patañjali in mahābhāṣya.

[3]indriyānām svaviṣayeśu anādiḥ yogyatā yathā |
anādiḥ arthaiḥ śabdānām sambandho yogyatā tathā ||(vākyapadīyam III 3.29)

naiyāyikas however are more realistic. They do not accept this as a natural relation. According to them, this relation is not 'natural' or 'siddha' but established by somebody. Naiyāyika's classify the relation into two types: primary and secondary. The primary relation further is of two types – a) saṁketa or īśvarecchā[4] and b) paribhāṣā. For example, the relation between a word 'cow' and an object cow or the relation between a technical term such as a 'circle' and a real object circle are primary relations. Of these, the first one is termed as a saṁketa, since nobody knows how this relation came into existence. In other words they are arbitrary. On the other hand, the technical terms are coined by somebody and hence the relation of a technical term to its object is termed as a paribhāṣā. To account for the metaphorical usages of a word, naiyāyikas admit a secondary relation, also known as lakṣaṇā.

It is the delimiter(avacchedaka) of this relation(śakti) - śakyatāvacchedaka, which causes a proper understanding of the meaning of a word. Śakyatāvacchedaka is also known as 'pravṛtti-nimitta' - the reason(nimitta) that warrants the use(pravṛtti) of a word to specify the desired meaning[5].

8.2 Example

The concept of Pravṛtti-nimitta helps one in understanding the underlying 'unity' among the seemingly different meanings of a word, and thus accounts for the polysemy. Let us take an example of the word *hari* in Sanskrit. Amarakośa lists 16 different meanings of this word (See fig 8.1).

Then how do we account for these seemingly different meanings? The word *hari* is

[4]Earlier naiyāyikas considered this relation to be īśvarecchā, but later naiyāyikas termed it as a saṁketa

[5]pravṛutteḥ śabdānām artha-bodhana-śakteḥ nimittam prayojakam (Nyāya kośa, pp 580)

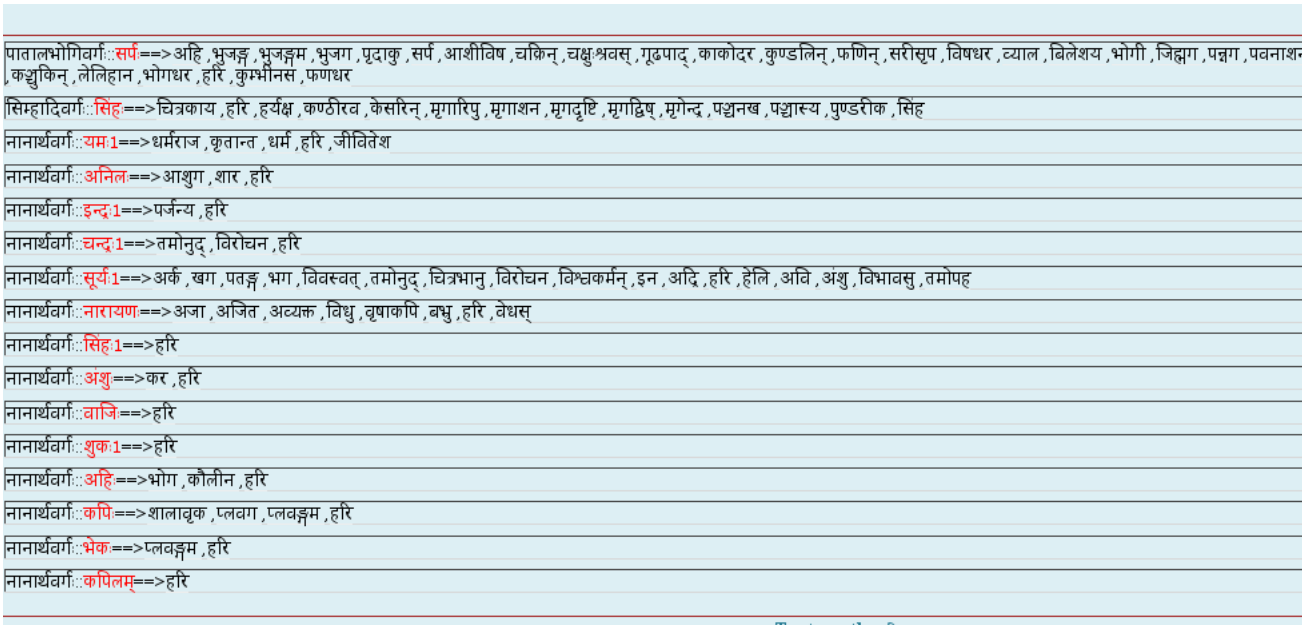


Figure 8.1: amarakosha entry

derived from the verbal root *hṛ* which means *haraṇe*(to steal). Thus according to this derivation one who steals is *hari*. And now everything falls in place. Viṣṇu is called hari because he takes away (steals) the sins of his devotees, monkey is called hari because it steals the eatables, a lion is called hari because he takes away one's life, and so on. Proper use of this pravṛtti-nimitta also gives a scope for creative use of such words.

8.3 The 'core meaning' in Modern Linguistics

In modern linguistics, though we find abstraction at the level of *form* (in terms of phonemes and morphemes, etc), abstractions at the level of semantics are very rare. Words are typically treated as polysemous and thus various meanings are just listed in the dictionary. This approach fails to give a holistic view of the meanings of a word. There are few exceptions, however [78].

"A monosemic approach, on the other hand, posits only one abstract sense for a word, and the various contextual meanings or readings that the word receives are predictable from features of the context in which the word occurs [82]"
For example, Saxton gives various senses of *just* as below.

- I live *just* around the corner. (in the sense of immediacy)
- I *just* got home from school. (in the sense of recently)
- Tina looks *just* like her mother. (in the sense of exactness)
- Your beet salad is *just* delicious. (in the sense of very)
- May be it is *just* hormonal. (in the sense of only)
- I can *just* squeeze through that fence. (in the sense of barely)

Saxton, by postulating the "core" meaning of *just* as *without deviating*, extracts the appropriate meanings from the features in the context. This concept of monosemic approach, thus, comes very close to the concept of Pravṛtti-nimitta.

Of late there have been attempts by the dictionary makers also to show how various meanings of a word are related, thereby, giving a holistic view of the various meanings a word expresses(see appendix D). Such an explanatory note helps a secondary language learner to acquire the 'seemingly diverse meanings of a word' with less effort.

8.4 Relevance of pravṛtti-nimitta in developing a Language Accessor

In anusaaraka, the first layer provides the śabdasūtra of each word. This śabdasūtra is developed on the basis of the pravṛtti-nimitta of the source language word. It is not an easy task to decide the 'core' meaning of a polysemous word. Though many times the etymology helps, factors such as diachronic changes, borrowing of words from other languages - sometimes leading to homonymy, make it difficult to discover the 'core' meaning. The task becomes more difficult when the 'core' meaning is to be expressed through other language. This then calls for a mechanism to handle the differences between two languages with regards to the packaging and labeling of concepts at lexical level, as discussed in chapter 3. These differences lead to five different combinatorial possibilities. We show in what follows[6] the solution anusaaraka provides in each of these cases, except the one-to-one mapping, of course.

8.4.1 Lexical Gap

The lexical gap may be present either at a concept-word level or at the function-word level. If it is at the level of a concept-word, we borrow the word as it is, till an equivalent in the target language is defined / coined and is in usage. For example the technical terms such as 'digital electronics', are borrowed into Hindi as it is, and displayed in the Devanāgarī script. The persons who read texts with such words are either familiar with the terms, or the texts they read define and explain these technical terms. And thus no more explanation of these terms is necessary.
Here is an example of an introduction of a technical term.

```
These satellites have devices which can receive signals from an earth
```

[6]At the risk of repetition, in order to provide a complete picture at one place, we may repeat some of the necessary discussions and examples from chapter 3 here.

```
station and transmits them back in various directions. Such a device
is called a TRANSPONDER.\\
```

In the anusaaraka output of such sentences, the word TRANSPONDER, for which there is a lexical gap in Hindi, will be borrowed into Hindi and transliterated into Devanāgarī script in the output. But if such a word represents some concept which needs cultural background, then anusaaraka interface pops-up a footnote providing necesary description.

If such a word is a function-word, then either we borrow the word, or use special markers and provide a help on the function it serves. For example, there is no functional equivalent to the English determiners in Hindi. So we borrow them and also provide a note on their usage. But in case of Telugu adverbial 'gā', instead of borrowing the suffix, we put a mark '*' as its Hindi equivalent, and provide an online help (see appendix C).

8.4.2 Many-One mapping

One may say many-one mappings do not pose any problem as far as the translation is concerned. It is not so. The many-one mapping may lead to either mis-interpretation or an ambiguity. For example, as is already noted in chapter 3, the following English sentence leads to ambiguity when translated into Hindi.

Eng: He gave a flower to her. (1)
Gloss: usane diyā eka phūla usako. (1a)
Hnd: usane usako eka phūla diyā. (1b)

From the Hindi translation it is not clear who gave a flower to whom, whereas in English it is clear that it is the male person who gave a flower to the female person. The source of ambiguity in Hindi is the third person pronoun because Hindi does not

English	Hindi
He	vaha{pu.}
She	vaha{strī.}
It	vaha{na.}
That	vaha-

Table 8.1: Mapping of third person pronouns in English into Hindi

mark third person pronouns for gender. The verb agreement in Hindi gives the gender information. Thus the two languages differ in 'where' the information is coded. Since the parsers are not reliable and anusaaraka claims 100% information preservation, anusaaraka proposes mappings for the pronouns from English into Hindi as shown in table 8.4.2.

Thus the anusaaraka output for the above English sentence would be
anu: usane{pu.} diyā eka phūla usako{strī.}.

If we describe the meaning of a word by a two dimensional region on a plane, then the union of 4 non-overlapping regions corresponding to the words *he*, *she*, *it* and *that*[7] of English correspond to a region *vaha* of Hindi. Using the nyāya terminology, the meaning of 'He' may be described as 'puṃliṅgatvāvacchinna vaha'(third person pronoun, delimited by masculine-ness), which we denote by *vaha{pu.}*.

8.4.3 One-Many mapping

When a word in English maps to more than one word in Hindi, we either try to express the underlying concept in words or represent all the disjoint meanings by '^',

[7]the dash sign after vaha in the meaning of that is used to indicate that it is an adjective in this context.

as follows:

Uncle: māmā^cācā^phūphā^mausā

cousin: bhāī^bahana{asahodara}

In the first example, we have listed all possible meanings joined by '^', to indicate that *uncle* may mean any of these. In the second case, we could have listed out all possible relation names such as cacerā(paternal), mamerā(maternal), phūpherā(of husband of father's sister), etc. denoting its meaning. But since we have a word *asahodara* which means *not born of the same womb*, *asahodara bhāī athavā bahana* will be an exact equivalent of *cousin*. But since *asahodara* is an adjective, and is not used in normal conversation in Hindi, we provide the equivalent of *cousine* as *bhāī^bahana* with *asahodara* within {} providing an extra information. Here the meaning of English word is an union of more than one non-overlapping regions in Hindi. The information within {} serves as an extra information which delimits the meaning. In nyāya terminology, then, the meaning of the English word *cousin* may be described as *asahodaratvāvacchinna bhāī_yā_bahana*.

8.4.4 Overlapped regions: A real challenge to develop śabdasūtra

Real challenging are those cases where one can not express a region in English as a union or intersections of various regions in Hindi, but needs much more complex sets of operations.

Let us look at the English word 'light'.

The dictionary entries for the word 'light' from the online English-Hindi dictionary[8] are:

light **Adj.** 1. halakā-

Example: This suitcase is light and good.

[8]available at http://ltrc.iiit.net/onlineServices/Dictionaries/Dict_Frame.html.

light **N.** 1. prakāśa

Example: I could see light in the room.

light **V.**

– 1. jalānā

Example: Mohan lighted the match-stick.

– 2. sulagānā

Example: Mohan lighted the cigarette.

– 3. prakāśita karanā

Example: The torch lighted the way for him.

– 4. prasanna honā

Example: His face lighted up when he heard the news.

The meaning of 'light' as an adjective is totally different from the meanings which correspond to the nominal and verbal usages of it. Thus this is a clear case of homonymy. It is obvious from the examples that the meanings of *light* as a noun and as a verb are related. But still, Hindi reader will definitely be puzzled by the range of usages of *light* as above and will fail to see the underlying 'thread' that connects these meanings. Here is an attempt to develop a thread running through different meanings of the word *light*.

It is very important to note that we are neither claiming to give a logical justification of why one may use the word 'light' in all the above senses, which for a Hindi speaker look as totally unrelated at the first instance, nor are we developing any theory for how the usage of a word and thereby its meaning gets shrinked or widened diachronically.

We are looking at the English usages and trying to establish a thread, which might be an artificial one, which may not reflect the correct historical developments, but which helps a Hindi reader to get a handle to relate seemingly different meanings and remind him at an appropriate time so that he can understand the English text correctly.

In the first example, *to light* may mean an activity that results in producing a *light*. According to vaiyākaraṇas a verbal root(dhātu) represents an activity(vyāpāra) and a resultant of the action(phala). Typically when the sense of a verbal root gets extended, the extension covers the similarity of an action or the similarity of the resultant. For example, lighting of the cigarette does not result in the production of light, but the underlying activity(of using the match stick to produce the flame and using it to light the cigarette) is same as that of lighting of the lamp. Lighting of a torch may not have the same underlying activity, but the result of the action is same, that is, it produces light. Similarly the presence of light illuminates the path and hence one can understand the usage *to light the way*. Finally the *merriment or spark* on the face may be explained as the metaphorical use of light.

So the **underlying thread** may be explained as

prakāśa ⇒ prakāśita karanā ⇒ prajvalita karanā ⇒ prakāśamāna karanā ⇒ jalānā (2)

Having arrived at the **thread** capturing all the senses, now we try to capture its essense in the form of a **concise expression** as

prakāśa [* karanā]/halakā (3)

The '*' indicates that prakāśa may take optionally some derived suffix. Note the placement of halakā as the second alternative and with '/' to mark the homonymy.

This concise expression[9] in (3) as well as the thread[10] in (2) is called a ***śabdasūtra***.

Thus a śabdasūtra stands for both a concise formula to represent the meaning and also a thread showing the connections between various senses of the meaning, which are apparently unrelated from the target language speaker's point of view. Around 1000 śabdasūtras are developed for high frequency English polysemous words.

Sometimes, the ambiguity is only at the word level, and not at the base (prātipadika) level. For example the word *leaves* has two analyses:
leaves : leaf, n, pl, and
leaves : leave, v, 3, sg
In such cases, we provide a 'sūtra' at the word level, such as
leaves = pattā{ba.}/choḍa
The '/' indicates that the meanings are unrelated.

8.5 Guidelines for developing śabdasūtra

The step by step procedure for developing the *śabdasūtra* is given below.

- Collect different senses of the word for which śabdasūtra is to be developed. Various monolingual and bilingual dictionaries, various electronic resources such as WordNet, etc. may be used for this purpose.
- Collect at least one example sentence for each of the described senses.
- Translate the sentences into the target language.

[9] alpākṣharam asandigdham sāravat vishvatomukham |
astobham anavadyam ca *sūtra*m *sūtra*vido viduḥ ||

[10] is called a sūtra in Sanskrit

- Try to minimise the number of target language equivalents to the maximum possible extent.
- Develop a thread linking all these minimised target language equivalents.
- Describe the connections in the thread.
- Develop a concise expression, which serves as an handle to refer to all these senses.

Various notations used in developing the concise formula are provided in the appendix B. The appendix E and F contain śabdasūtra for the words 'as' and 'case' respectively.

8.5.1 Śabdasūtra and fidelity

Now we come back to the original question: In anusaaraka how do we ensure that a Hindi reader has complete 'access' to the English text?

The śabdasūtra in the form of a concise formula is used in the first layer of the anusaaraka output. Every word is split into a base and a suffix (prakṛti and pratyaya). We provide the meaning of each of these as a śabdasūtra. The underlying assumption is that the meaning of a word can be composed from its constituents. In case the meaning is non compositional, we provide a śabdasūtra for the complete word. So the first time reader will have to go through the threads understanding the explanation. In the user interface therefore,

- We provide the śabdasūtra for each word.
- We also provide additional help necessary to understand the sūtra. This help consists of

- Dictionary entry that lists various meanings of a word,
- An example English sentence for each meaning, with its Hindi translation,
- A small essay showing how the various meanings are related, and
- The śabdasūtra which ties all these meanings concisely in the form of a formula.

8.6 Śābdabodha in anusaaraka

The process of śābdabodha describes the steps involved in 'understanding' the meaning of a sentence after listening to it. Anusaaraka produces the output in several layers. We illustrate below with examples from English-Hindi anusaaraka, how the layers produce the śābdabodha of the English sentence.

- Example 1:

  Eng: Rats kill cats.
  anu-Hnd: cūhā{ba.} māra{0} billī{ba.}
  An anusaaraka reader who has undergone the training on English grammar from Hindi viewpoint, would remember about the special vṛtti, viz. _V_, i.e., a transitive verb requires two arguments, whose positions are fixed, and are to its immediate left and immediate right. The argument on the left is the subject and the one on the right is the object, and that the subject is *abhihita*. With this knowledge, then he establishes the relations between different words as, *cūhā* is the abhihita, the verb is in active voice, and thus now appealing to the knowledge of Hindi grammar, the abhihita should be kartā and hence should take nominative case, and the other word, viz. *billī* should be the karma, and hence should take accusative case. And thus a anusaaraka reader understands the English sentence through Hindi as
  Hnd: cūhe mārate_haiṃ billiyom_ko.

- Example:2

  Eng: Shankar was returning home on his bicycle after a football match.
  The first layer output of this sentence, where each word is split into a stem and an suffix, and their meanings are provided is shown below.
  anu-Hnd: Shankar thā vāpasa_lauṭa2 {ing} ghara on{→ para} vaha{pu.}_kā sāikila after{→ ke_bāda[pīche]} a phuṭabāla maicaˆjoḍa.

  The '2' in the meaning of the verb *return* indicates that the verb can be both intransitive as well as transitive. Accordingly, a Hindi reader has to choose the correct alternative between *lauṭa* and *lauṭā*. The → indicates that the preposition should not be read 'in situ', but should be placed at an appropriate position to the right. Thus a user has a heavy load in interpreting the meaning of this sentence:

  – Deciding the correct meanings of the words *return*, *after*, *match*, and the verbal suffix *ing*, and
  – Deciding the preposition phrase boundary.

  At this stage the reader may look at the second layer output of the same sentence, to reduce the burden of interpretation (with a possible cost of misinterpretation).

  anu-hin-layer2: Shankar — vāpasa_lauṭa2{0_rahā_thā} ghara on{→ para} vaha{pu.}_kā sāikila after{→ ke_bāda[pīche]} a — phuṭabāla_maica.

  At this stage, the words 'are returning' and 'football match' are groupd together. Hence the burden of disambiguation of the word *match* and the suffix *ing* is reduced at this stage. The expectancy now comes into play and helps in deciding the meaning of the verb *return*. The verb *return* as an intransitive

verb (in the sense of lauṭa) does not have an expectancy of an object, but as a transitive or ditransitive verb(in the sense of lauṭā) has expectancy of a direct object. The absence of a direct object then leads to the choice of lauṭa.

Yogyatā(competency) helps in the proper choice of meaning of the preposition *after* viz. *ke_bāda*. *Ke_pīche* is used when the noun specifies a location in the space whereas, when the noun indicates an event or time, then the meaning is *ke_bāda*. The reader on the basis of football match, which is an event, therefore chooses *ke_bāda* for *after*. Yogyatā also helps further to decide the preposition boundaries of the preposition *on*.

8.7 Conclusion

In this chapter we have seen the innovative use of the concept of pravṛtti-nimitta to represent the meaning of a word in the target language. The concept of śabdasūtra is evolved as a means to express the pravṛtti-nimitta in a compact form. We have further shown with examples, how the śabdasūtra together with the notion of ākāṅkṣā and yogyatā help in getting the meaning of the sentence. Assuming that the reader of the target language shares the world knowledge, domain knowledge etc. with the source language reader, and taking into consideration the incompatibility between the source language and the target language in coding the information at various levels, the notion of śabdasūtra plays an important role in the formulation of image of source language words in the target langauge.

Chapter 9

Pāṇinian Interface for English Parsers

9.1 Introduction

Last decade has seen introduction of several parsers for English ranging from rule based to statistical. Within rule based again one sees parsers with a wide variety of formalisms such as Minipar [59] based on minimalism, Enju parser [94] and LKB parser [29] based on HPSG, link parser [86] based on dependency grammar, XTAG parser [46] based on Tree Adjoining Grammar, to name a few. There are around half a dozen statistical parsers for English viz. Collins [27], Charniak [24], Stanford parser [61], re-ranking parser [25] and so on. The native output of all these parsers is naturally the grammar formalisms they follow. In case of rule based parsers, it is the grammar formalism they are based on, and in case of statistical parsers, it is the Phrase structure trees, since they are trained on the Penn Treebank which is annotated using Phrase Structure Grammar (PSG).

Recent years has also seen a growing trend towards producing dependency output in addition to the constituency trees. The dependency format is preferred over the

constituency not only from the evaluation point of view [59] but also because of its suitability [62] for a wide range of NLP tasks such as Machine Translation (MT), information extraction, question answering etc. However no two dependency output formats match with each other. There is no consensus among the dependency parser developers on the number of dependency relations and names of these relations.

Paninian Grammar (PG), the first dependency formalism, though is developed specifically for Sanskrit, has potential to provide guidelines for producing the dependency output of English sentences. Such guidelines will also be helpful in developing interfaces for the existing parsers so that one can plugin different parsers to the existing Machine Translation software.

We first summarize the issues involved with reference to English language parsing based on the dependency format output of the current English parsers. In the third section we highlight the Information theoretic viewpoint of PG, with special emphasis on English language. Fourth section contains guidelines for producing the dependency output for English, in the light of PG.

9.2 Dependency format output: some issues related to English

A dependency relation is an asymmetric binary relation mapping a modifier(or dependent) to the modified(or governor). The word being modified is the head. A word may have several modifiers but can modify only one word. If there are n words in a sentence, n-1 relations are necessary and sufficient to describe the parsed output.

There is a very close relationship between the dependency grammar and the link grammar [86] on which is based the link parser. The relations in link parser, however, are

not directional. The number of relations used in link parser is 106. Minipar also produces dependency format output and uses 59 relations. Carroll [23] and King [50] have proposed a set of dependency relations. Marneffe et al [62] have suggested modifications to these relations, largely based on practical considerations. The number of relations proposed by Marneffe are 47. Thus we see that there is a lot of variation among different parsed outputs with respect to the number of relations.

We looked at parsed outputs of different parsers for a wide range of sentences and recorded the phenomena where the parsed outputs differ. We also noticed certain cases where none of the parsers' performance was acceptable. The differences in their performance could be related to the issues summarized below.

a) Whether to treat function words such as prepositions, auxiliary verbs, etc. as words indicating relations thereby avoiding relations between these words with other content words or to treat these words at par with the content words?

This will have serious effect on the number of content words and the number of relations in a sentence.

b) The basic assumption of dependency grammar is that a modifier modifies only one word. In the following sentence

Eng: Ram went home and slept. (1)

Ram is a modifier of went as well as slept. Whether the parser should produce both the relations or only one?

Similarly in the sentences with missing wh -relativizer

Eng: I saw the man you love. (2)

The snake the mongoose attacked hissed loudly. (3)

whether the output should account for the missing wh-relativizer?

In case of subject and object control verbs such as

Eng: Ram persuaded Mohan to study well. (4)

Eng: Ram promised Mohan to study well. (5)

should the output account for the sharing of semantic roles by verbs?

c) What should be the level of analysis – syntactic (specifying the subject, object relations), semantic (specifying the thematic roles), or something else?

d) Should the heads be decided semantically or syntactically? For example, in case of *a cup of tea*, the semantic head is *tea*, whereas the syntactic head is *cup*. In case of *growth of industry*, *growth* is both the semantic as well as syntactic head.

e) Should the sentences

Eng: Ram is good. (6)

and

Eng: Ram is a doctor. (7)

be treated alike, with semantic representation as *good(Ram)*, and *doctor(Ram)* respectively, or should they be analyzed differently, reflecting different underlying phrase structures?

To answer these questions, we look at English language from the 'information coding' point of view. We seek answers for the following questions.

i) What means does English use to code the information about relations?

ii) What is the manner of coding the information, and finally,

iii) What is the semantic content of these relations?

9.3 Pāṇinian Grammar

According to Pāṇinian Grammar (PG), a modifier may be classified into two major categories: samānādhikaraṇa (modifier and modified having the same locus), and vyadhikaraṇa (modifier and modified have different loci).
Examples of samānādhikaraṇa modifiers are

1. a determiner modifying a noun (the boy)
2. an adjective modifying a noun (good boy)

Examples of vyadhikaraṇa modifiers are

1. nominal expressions modifying a verbal root, also known as the kāraka relations,
2. a verb modifying another verb, etc.

Essentially, the samānādhikaraṇa modifier and the corresponding modified head denote the same thing, and belong to the same word group[1]. So this kind of relation is a 'word-group-internal' relation. On the other hand the vyadhikaraṇa modifier and the corresponding modified head belong to different word groups, and hence the relation involved here is 'across-the-word-group' relation. In short, the vyadhikaraṇa modifiers are the building blocks of the parsed structure, whereas, with the samānādhikaraṇa modifiers, the modifiers add flesh to this structure.

The most important vyadhikaraṇa modifiers are the *kāraka* relations.

[1]Of course, there are cases where the words may belong to different word groups and still may have the same locus, as in the case of *He is a doctor*.

1. In Bharati et al [10] it has been pointed out that English codes the kāraka relations in position as well as through prepositions.

2. Languages do not code all the kāraka relations explicitly. For example, when a word has more than one kāraka role with respect to different verbs in the surface structure of a sentence, only one kāraka relation is coded and other kāraka relation needs to be inferred from the language's grammatical rules (language conventions) or through the properties of lexical items. For example in sentence (1), it is the language convention which tells *Ram* is the subject of both the verbs *went* and *slept*. In sentences (2) and (3), it is the syntax of English which allows wh-drop and thereby allow sharing of more than one kāraka role by the same nominal expression. In sentence (5) the information that subject of *study* is *Ram*, and in sentence (4) it is *Mohan*, which is coded in the meaning of lexical items *promise* and *persuade* respectively.

3. According to PG, the kāraka relations are the relations which map nominal expressions to verbal roots. These are syntactico-semantic relations. These indicate the optimum semantic analysis one can do using the language string and the language conventions alone without appealing to the world knowledge[2]. Given the fact that present day computers are still not capable of handling the world knowledge, from computational point of view, it is a major milestone in the language analysis. One kāraka relation may correspond to more than one thematic role. For example, in the following sentences

   Eng: Ram opened the lock with this key. (8)

   Eng: This key opened the lock. (9)

   Eng: The lock opened. (10)

   Ram, this key and the lock are all kartā, whereas their thematic roles are viz.

[2]See chapter 5 for more details.

agent, instrument and goal respectively. Similarly each semantic role may get realized into more than one kāraka relation. For example, *key* in sentence (8) is karaṇa kāraka and in sentence (9) karta kāraka. *Lock* is the karma kāraka in sentences (8) and (9), whereas karta kāraka in sentence (10).

To summarize,

1. English codes the kāraka relations both by position as well as through the prepositions.

2. Some relations are coded explicitly and some implicitly.

3. The maximum semantics one can extract is the syntactico-semantic relations and not the thematic roles.

9.4 Guidelines for producing dependency output for English

We answer the issues raised in the second section, which will lead to the guidelines for producing the dependency output for English. Appendix G and Appendix H contain the outputs generated by the Stanford and Link parsers respectively, while the appendix I contains the dependency trees following Pāṇinian grammar.

1. In the light of earlier discussion, it is clear that we treat the prepositions connecting a noun with a verb or another noun as a relation rather than a content word. Further the auxiliary verbs together with the main verb form a 'semantic unit' leading to a word group with main verb as the head. Hence the auxiliary verbs should be grouped with the main verb, and there is no necessity of mentioning the internal relations.

2. Sentences (1) through (5) are all examples of kāraka sharing and implicit encoding of the unspecified kāraka relations. The implicit encodings are typically language grammar and lexicon specific and hence need to be made explicit in the parsed output.

3. On the basis of the discussion above, it is clear that, language codes only syntactico-semantic relations. So what one can extract from the language string alone is only syntactico-semantic relations and not the thematic roles. For Sanskrit we have kāraka - vibhakti mapping rules described in the Aṣṭādhyāyī. Similar rules need to be worked out for English. Till then, we describe the relations in terms of objects of prepositions or by subject and object positions. In other words, the relations will therefore be marked as subject and object, in case they are expressed by position, and by the preposition-object, such as by-obj, with-obj etc., in case they are expressed by prepositions.

4. In case of 'of', since it is the syntactic head expressed by the relation, we mark only the syntactic head. The determination of semantic head requires the world knowledege, and hence should be dealt with in a separate module.

5. In English two sentences may have different Phrase structures, but their semantic content may be the same. PG treats them in a uniform way, by postulating a samānādhikaraṇa relation between Ram and good, and also between Ram and doctor. This in fact is an example of samānādhikaraṇa modifier across the word groups!

9.4.1 Conclusion

In the light of above discussion the relations may be classified into three categories viz. word-group-internal relations, across-word-group-explicitly marked relations, and across-word-group-implicitly marked relations. The word-group-internal rela-

tions may be best handled by the constituency trees, whereas the across-word-group relations may best be handled by the dependency relations. Chunkers may be the reliable tools for marking the inter-word-grouping. The word grouper developed in-house performs better than the chunker on main verb-auxiliary verb grouping. Handling the implicit relations involve some heuristic rules. These need to be, therefore, marked separately.

We have shown that various parsers differ in their behaviour with respect to the issues raised above. For example Link parser treats prepositions as content words. It also treats sentences (6) and (7) differently. Stanford parser and Enju parser on the other hand try to do deeper semantic analysis leading to over-generalizations in some cases. The differences among these parsers make it difficult to compare the parsers qualitatively.

Interfaces based on the principles outlined above are being developed for various parsers. These interfaces are easy to use by a layman for understanding the 'parsed output' without much linguistic training [16]. It also facilitates comparison of different parsers. The initial motivation for building such interfaces was to provide a plugin facility for plugging-in different parsers.

Chapter 10

Conclusion

There is a radical shift proposed in the architecture of the anusaaraka from its earlier version. The earlier version, very well demonstrated its usefulness as an aid to overcome the language barrier. But the output of the system involved a learning component. In this thesis, we showed, how the system can be improved further, taking it towards Machine Translation, still retaining the fidelity to the original text. We have also shown how the various concepts from IGT are relevant in developing a Machine Translation system. We discuss below some experimental feedback we got on the present system which provides directions for future development.

10.1 Present: some experimental feedback

Anusaaraka developed with an aim to provide a faithful access to the SL text, was found to be a good tool to assist the students learning English as a second language too. The usefulness and effectiveness of the tool was tested at various Hindi medium schools of Madhya Pradesh, India. Madhya Pradesh Sanskrit Board at Bhopal provided help from time to time to get feedback on the anusaaraka at various stages of its development.

The first three experiments were carried out with only first layer of anusaaraka output [13]. Thus students had to undergo some training in the śabdasūtra and also the lessons describing the structural differences in English and Hindi. The students who, prior to the experiment, were not able to comprehend simple sentences were at the end of a week's period able to comprehend stories of 10-15 sentences, with the help of anusaaraka output. Most of these stories were from Children's story books. During these experiments need was felt to provide more help in the form of word groupings, phrase boundaries, likely POS tags, etc., which resulted in the present architecture.

The main emphasis of anusaaraka being the faithfulness, scientific texts and other texts carrying 'information' are more relevant for the use of anusaaraka rather than texts meant for amusement, entertainment, etc. To find out how useful the anusaaraka in current state (as of 2005) is, it was decided to conduct some more experiments.

With the improved architecture, again an experiment was planned, but this time with the science texts. The students of 10^{th} class who have studied 9^{th} standard science texts through Hindi medium were given the anusaaraka outputs of science texts, in order to know how much this tool helps a student to understand the scientific English texts with which s/he is already familiar with.
The following are our observations:

- The major problem was with the scientific concepts. Though the students had studied the scientific texts through Hindi medium, they found it difficult to comprehend the anusaaraka output of the same text from English into Hindi because, they were not thorough with the scientific concepts involved.

- Second problem was the domain specific constructions. It was necessary to develop domain specific modules to handle the specific syntactic constructions and also use domain specific dictionary to get better results.

- Finally it was noticed that adding a WSD module will improve the quality and reduce the burden on the user further to a great extent.

This then defined the course of our future work.

10.2 Future

The first version of the anusaaraka demonstrated the technological feasibility of anusaaraka to overcome the language barrier. The second version further demonstrated the relevance of Indian Grammatical Tradition in the development of this technology. The design of architecture showed various ways to add comfort to the end product and make the end product more user friendly and comfortable reducing the burden on the user.

As it was noticed during the experiments in the schools, now, in the third version a strong need is felt about the Word Sense Disambiguation module. Use of WASP workbench helped us in developing the WSD rules semi-automatically. But the rules were so large in number that, it failed to provide any linguistic insight. There is a need to keep the rules in human readable form, so that, if need arises, one can improve upon them. Secondly, the task being voluminous, with always scope for further improvement, there is a need to involve large number of people. Work is in progress towards developing a user friendly interface for developing WSD rules. The importance of involvement of the society in the development of Oxford dictionary, and online knowledge repositories such as Wikipedia are well-known. The goal of the next version of anusaaraka then should be to make the system self sustainable or in other words, provide an opportunity to the users for participation in the development makes a consumer a contributor.

The close interaction of Sanskrit scholars with the system, should provide better in-

sights in developing parser interfaces for various parsers, by the way of designing an intermediate parser representation based on Pāṇinian system.

The architecture together with an expert shell for decision making should provide a good base for Linguist's Workbench, where a user can observe various language behaviours, play with various parsers, understand different means of coding information in language strings, understand the information dynamics within a language, and understand the information flow from one language into the other in an MT system.

Appendix A

Pos-Voting

1.A	POS
1.B	GENIA
1.C	Hepple
1.D	GROK
1.E	Brill
1.F	Stanford
1.G	VOTING

3.1.A	One	day	when	I	was	nine	,	my	cousin	Mourad	,	who	was	considered	crazy	by	everybody	except	me	,	came	to	my	house	.
3.1.B	CD	NN	WRB	PRP	VBD	CD	,	PRP$	NN	NNP	,	WP	VBD	VBN	JJ	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.
3.1.C	CD	NN	WRB	PRP	VBD	CD	,	PRP$	NN	NNP	,	WP	VBD	VBN	JJ	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.
3.1.D	CD	NN	WRB	PRP	VBD	RB**	,	PRP$	NN	NNP	,	WP	VBD	VBN	NN**	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.
3.1.E	CD	NN	WRB	PRP	VBD	CD	,	PRP$	NN	NNP	,	WP	VBD	VBN	JJ	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.
3.1.F	CD	NN	WRB	PRP	VBD	CD	,	PRP$	NN	NNP	,	WP	VBD	VBN	JJ	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.
3.1.G	CD	NN	WRB	PRP	VBD	CD	,	PRP$	NN	NNP	,	WP	VBD	VBN	JJ	IN	NN	IN	PRP	,	VBD	TO	PRP$	NN	.

8.1.A	It	was	not	morning	yet	but	it	was	summer	and	it	was	light	enough	for	me	to	know	I	was	not	dreaming	.
8.1.B	PRP	VBD	RB	NN	RB	CC	PRP	VBD	NN	CC	PRP	VBD	JJ	RB	IN	PRP	TO	VB	PRP	VBD	RB	VBG	.
8.1.C	PRP	VBD	RB	NN	RB	CC	PRP	VBD	NN	CC	PRP	VBD	JJ	RB	IN	PRP	TO	VB	PRP	VBD	RB	VBG	.
8.1.D	PRP	VBD	RB	JJ**	NN**	CC	PRP	VBD	NN	CC	PRP	VBD	JJ	NN**	IN	PRP	TO	VB	PRP	VBD	RB	JJ**	.
8.1.E	PRP	VBD	RB	NN	RB	CC	PRP	VBD	NN	CC	PRP	VBD	NN**	RB	IN	PRP	TO	VB	PRP	VBD	RB	VBG	.
8.1.F	PRP	VBD	RB	NN	RB	CC	PRP	VBD	NN	CC	PRP	VBD	JJ	JJ**	IN	PRP	TO	VB	PRP	VBD	RB	VBG	.
8.1.G	PRP	VBD	RB	NN	RB	CC	PRP	VBD	NN	CC	PRP	VBD	JJ	RB	IN	PRP	TO	VB	PRP	VBD	RB	VBG	.

Figure A.1: POS voting Result

Appendix B

Śabdasūtra Notation

notation:	/
illustration:	**a/b**
explanation:	a **or** b; We use / in case of homonymy.
example śabdasūtra:	lie: *jhūṭa bolanā/leṭanā*.
meaning of śabdasūtra:	lie: *jhūṭa bolanā* **or** *leṭanā*.
notation:	[ ]
illustration:	**a**[**b**]
explanation:	a **or** b; We use [ ] in case the meanings are related.
example śabdasūtra:	when: *jaba*[*kaba*].
meaning of śabdasūtra:	when: *jaba* or *kaba*.
notation:	˜ [ ]
illustration:	**a**˜ [**b**]
explanation:	a **or** a_b;
example śabdasūtra:	then: *taba*˜[*to*].
meaning of śabdasūtra:	then: *taba* **or** *taba*˜*to* .
notation:	[<]
illustration:	**a**[<**b**]
explanation:	Basic meaning is 'b', but the most frequent meaning is 'a', and is derived from b.
example śabdasūtra:	absorb: *sokhanā*[<*andara khīṃcanā*].
meaning of śabdasūtra:	Basic meaning of the English word *absorb* in Hindi is *andara khīṃcanā*; but the frequent meaning is *sokhanā* .

notation:	[>]
illustration:	**a**[>**b**]
explanation:	Basic meaning is 'a' and it is also the most frequent meaning, but 'b' is derived/extended from 'a'.
example śabdasūtra:	leave: *chodanā*[>*chuṭṭi*].
meaning of śabdasūtra:	Basic as well as the most frequent meaning of the English word *leave* is *choḍanā* in Hindi, and the noun meaning *chuṭṭi* gets derived from the verb.
notation:	‘
illustration:	**a‘**
explanation:	The meaning is almost 'a', but there is some difference which is indicated by the mark.
example śabdasūtra:	it:*vaha*‘.
meaning of śabdasūtra:	There are cases such as dummy it, barring those, the meaning of English *it* is *vaha* in Hindi.
notation:	^
illustration:	a^b
explanation:	The meaning is neither 'a' nor 'b', but something which is expressed by both 'a' and 'b' both.
example śabdasūtra:	grandmother: dādī^nānī.
meaning of śabdasūtra:	The word *grandmother* of English neither expresses the meaning of *dādī* alone nor that of *nānī* alone (in Hindi), but it expresses both.
notation:	**{}**
illustration:	**a{b}**
explanation:	meaning of a is restricted by the meaning of b.
example śabdasūtra:	she: vaha{strī}.
meaning of śabdasūtra:	*she*, in English, is a 3P pronoun marked for feminine.
notation:	*
illustration:	**nā***
explanation:	augment the meaning of nā by supplying appropriate suffixes/words in the context.
example śabdasūtra:	nā*
meaning of śabdasūtra:	the verbal nominaliser suffix *nā*, in Hindi, may take any of the possible postpositions such as ke_liye, se, ko, etc.

notation:	**[(-b)>]**
illustration:	**a[(-b)>k]**
explanation:	In the absence of 'b', the meaning is 'k', else it is 'a'.
example:	few: kucha[(-a)>na_ke_barābara].
meaning of śabdasūtra:	In the presence of *a* (as in *a few*), the meaning of the English word *few* in Hindi is *kucha*, else the meaning is 'na_ke_barābara'.

Appendix C

Telugu Adverbial Suffix -gā

Telugu adverbial suffix _*gā* has following meanings
_gā: sā/jaisā/ke_rūpa_meṁ

The Telugu suffix -gā changes an adjective to an adverb. Since Hindi does not have an equivalent suffix, to indicate the lexical gap, we replace it with * in the Hindi output.

Here are some examples illustrating the various usages of the adverbial suffix -gā of Telugu.

Tel: batta tellagā uṁdi.
Anu-Hin: kapaḍā sapheda_* hai{3_ pu_e}.
Hin: kapaḍā sapheda_sā hai.

Tel: bhārata deśāniki uttaraṁgā himālaya parvatālu unnāyi.
Anu-Hin: bhārata deśa_ko‘ uttara_* himālaya parvata hai{3_non-pu_ba}.
Hin: bhārata deśa_ke uttara_meṁ himālaya parvata haiṁ.

Tel: vādu toṁdaragā veḷḷāḍu.
Anu-Hin: vaha jaldī_* gayā{3_pu_e}.
Hin: vaha jaldī_se gayā.

Tel: nīvu eṁduku ālasyaṁgā vaccāvu?
Anu-Hin: tuma kyoṁ dera_* āyā{3_e}?
Hin: tuma kyoṁ derī_se āye?

Tel: ī lekka cālā gajibijigā uṁdi.
Anu-Hin: yaha- hisāba bahuta gaḍabaḍī_* hai{3_non-pu_e}.
Hin: yaha hisāba bahuta gaḍabaḍī hai.

Tel: ī nimmapaṁdu pullagā uṁdi.
Anu-Hin: yaha- niṁbū khattā_* hai.
Hin: yaha niṁbū khattā_sā hai.

Tel: vāḍu kopaṁgā unnāḍu.
Anu-Hin: vaha{pu.} gussā_* hai{3_pu._e.}
Hin: vaha gusse_meṁ hai.

Tel: vāri kutuṁbaṁ ārthikaṁgā venakabaḍiṁdi.
Anu-Hin: unakā kutuṁba ārthika_* pichaḍa_gayā.
Hin: unakā kutuṁba ārthika_dṛṣṭī_se pichaḍa_gayā.

Tel: vāru nāku taṁḍrigā unnāru.
Anu-Hin: vaha{honorific_ba.} mujhe‘ pitā_* hai{3_pu_bahu}.

Hin: ve mujhe pitā_samāna haiṁ.

Tel: narasimharāva pradhānamaṁtrigā pani cesādu.
Anu-Hin: narasimharāva pradhānamaṁtri_* kāma kiyā{3_pu_e}.
Hin: narasimharāva pradhānamaṁtri_ke_rūpa_meṁ kāma kiyā.

Tel: māṃtrikuḍu pillavāṇṇi kukkagā ceśāḍu.
Anu-Hin:jādūgara ne bacce_ko kuttā_* kiyā{3_pu_e}.
Hin:jādūgara ne bacce_ko kutte_ke_jaisā kiyā.

Tel: māṃtrikudu pillavāṇṇi kukkagā mārcādu.
Anu-Hin:jādūgara ne bacce_ko kutte_* badalā{3_pu_e}.
Hin:jādūgara ne bacce_ko kutte_ke_rūpa_meṁ badalā.

Below we give features which trigger various meanings in Hindi.

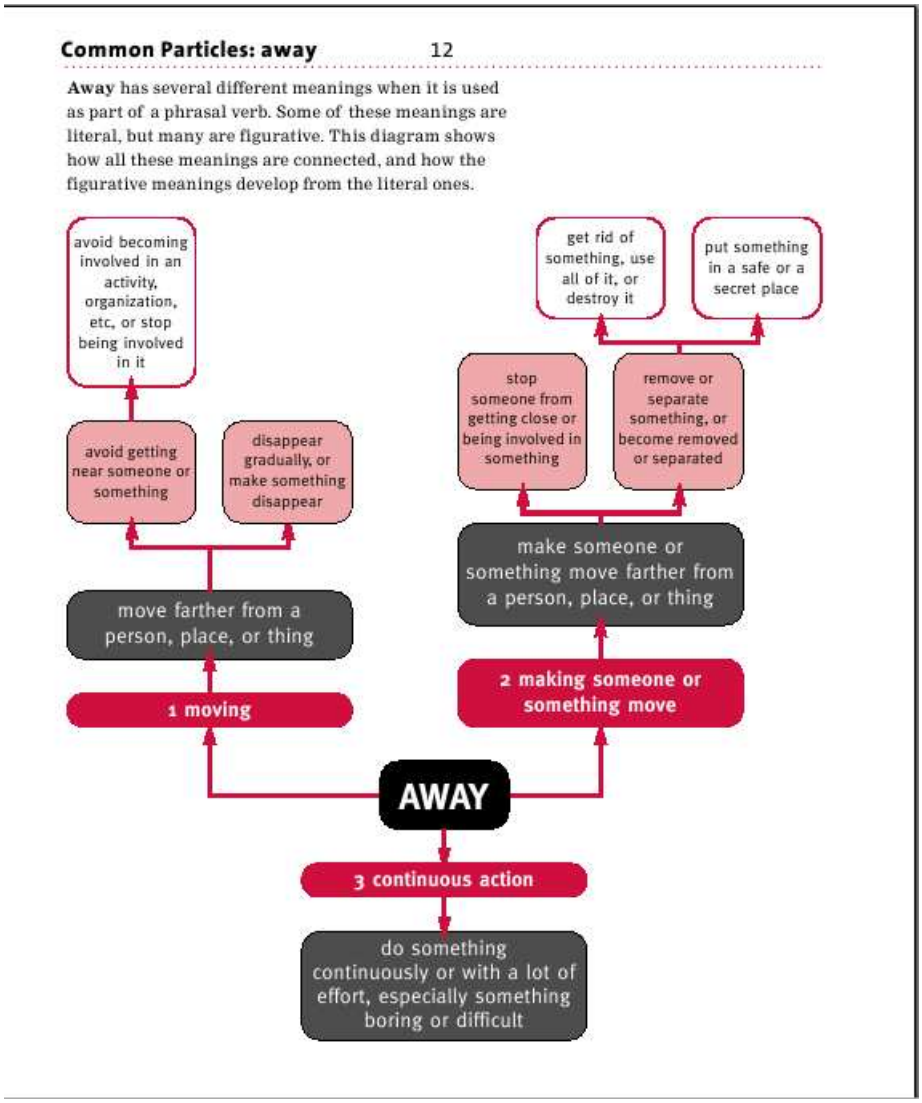
- gā + uṃdi -> sthiti_meṁ_hone_kī_viśeṣatā / meM hai
- gā + pani ceyyi -> ke_rūpa_meṁ kāma_karanā
- words indicating direction + gā -> se

Appendix D

Macmillan's Phrasal Dictionary: sample entry

The Macmillon Phrasal Verbs Plus[1] provides special entries on the 12 most common particles (away, back, down, out etc) explain how they contribute to the meaning of phrasal verbs. Here we produce a diagram explaining how the different senses of the phrasal verbs with particle 'away' emerge from the 'core meaning' of the word 'away'. Such a diagram helps a new learner to absorb the meaning quickly and also get a holistic view of the different connections.

[1] http://www.macmillandictionaries.com/about/PhrasalVerbs/phrasalverbs.htm


Common Particles: away 12
Away has several different meanings when it is used as part of a phrasal verb. Some of these meanings are literal, but many are figurative. This diagram shows how all these meanings are connected, and how the figurative meanings develop from the literal ones.
avoid becoming involved in an activity, organization, etc, or stop being involved in it
avoid getting near someone or something
disappear gradually, or make something disappear
move farther from a person, place, or thing
1 moving
get rid of something, use all of it, or destroy it
put something in a safe or a secret place
stop someone from getting close or being involved in something
remove or separate something, or become removed or separated
make someone or something move farther from a person, place, or thing
2 making someone or something move
AWAY
3 continuous action
do something continuously or with a lot of effort, especially something boring or difficult

Appendix E

Śabdasūtra for 'as'

Various senses of the English word *as* are shown below, with the appropriate Hindi meanings in the context. The sentences with similar Hindi meanings are grouped together.

- (*tulanā*) [*utanā*] ... *jitanā / ke jitanā*

  I like this jacket better than that one, but it costs twice as much.

  twice as much: *dugune kī jitanī*

  They live in the same town as my parents.

  same town as: *usī śahara meṁ jisa meṁ*

  I'd never seen him looking so miserable as he did that day.

  so miserable as: *utanā duḥkhī jitanā*

- (*sāmya*) *ke rūpa meṁ/kī taraha*

  He went to the fancy-dress party dressed as a banana.

  as a banana: *kele kī taraha*

  As a child, Mary had lived in India.

  As a child: *bacce ke rūpa meṃ → bacapana meṃ*

- (*sāmya*) *kī taraha / ke jaise*

  It could be used as evidence against him.

  as evidence: *kī taraha*

- (*tulanā*) *kāla → (kāraṇa) cūṁki*

  As it was getting late, I decided to book into a hotel.

  as : cūṁki

  You can go first as you're the oldest.

  as : cūṁki

- (*sāmya*) *kāla jaise / jaba*

  As I was getting into the car, I noticed a piece of paper on the floor.

  as: *jaise/jaba*

  He gets more attractive as he gets older.

  as: *jaise/jaba*

- (*sāmya*) *kāla → ke bāvajūda*

  Angry as[1] he was, he couldn't help smiling

  as: *ke bāvajūda*

- (*sāmya*) *vyavahāra → jaisā*

  Do as I say!

  as: jaisā

[1]Note that here the information is in position.
Compare it with
As he was angry, he shouted at her.
In this example, *as* is used in the sense of *because*.

When I arrived at the party, someone else was dressed exactly as I was.

as: *jaisā*

He got divorced, (just) as his parents had done years before.

as: *jaisā*

- (*sāmya*) *anurūpa* → *jaisā ki*

  As I thought, Danny was to blame.

  as: *jaisā ki*

Thus we observe that *as* basically is used when there is a comparison between two similar things. Hindi uses different words to express comparison at various levels such as behaviour, look, place, time, etc. From these examples, the meaning of *as* may be summarised as follows.

sāmya { *tulanā*} -|- *rūpa*
|- *vyavahāra*
|- *deśa*
|- *kāla* → *kāraṇa* → *ke bāvajūda*

which shows the thread running through various senses.

The concise formula may be:

as: sāmya / tulanā

Appendix F

Śabdasūtra for 'Case'

Look at the following sentences illustrating the uses of *case*, along with its Hindi meaning in each case.

1. It was obviously a **case** of blackmail. *māmalā*
2. Keep the ornaments in the jewllery **case**. *ḍibbā*
3. I need three pillow **cases**. *khola*
4. Objects in the museum have been displayed in big glass **cases**. *almārī*
5. Write your name in the upper **case**. *baḍe akṣara*

Other examples of the word case are as in compounds, such as 'hot-case, suit-case, brief-case'. Looking at these and the one in the second example, the meaning *ḍibbā* seems to be closer. But we know that English has another word 'box' for *ḍibbā*. We observe one speciality in these usage of case that, the boxes it refers to actually are the boxes specially designed according to the special requirement of the object for which it is meant for. The 'glass case' is also a special type of case which is designed to be suitable for the display of items in the museum. 'Pillow case' is also a special

type of cover that suits the shape of the pillow.

The Compact Oxford English dictionary[1] has two entries for the word 'case' as shown below.

- entry 1
  noun 1 an instance of a particular situation or set of circumstances. 2 an instance of a disease, injury, or problem. 3 an incident under official investigation by the police. 4 a legal action that is to be or has been decided in a court of law. 5 a set of facts or arguments supporting one side of a debate or lawsuit. 6 a person or their situation as a subject of medical or welfare attention. 7 Grammar an inflected form of a noun, adjective, or pronoun expressing the semantic relation of the word to other words in the sentence: the possessive case.

  ORIGIN Latin casus 'fall, occurrence, chance'.

- entry 2
  - noun 1 a container or protective covering. 2 Brit. a suitcase. 3 a box containing twelve bottles of wine or other drink, sold as a unit. 4 each of the two forms, capital or minuscule, in which a letter of the alphabet may be written or printed.
  - verb 1 enclose within a case. 2 informal reconnoitre (a place) before carrying out a robbery.

  ORIGIN Old French casse, from Latin capsa 'box, receptacle'; sense 4 derives from a container for holding type in printing; two cases were set on a stand, the higher one for capitals (upper case) and the lower for minuscule (lower case).

Thus this etymology captures the fifth sense also. Now regarding the first example above, according to Oxford's dictionary, is a separate entry, which originates from the Latin casus, meaning 'fall, occurrence, chance'.

So one may come up with a sūtra: *māmalā / ḍibbā viśseṣa*

[1] accessed in June 2009 from the web http://www.askoxford.com/dictionaries/compact_oed/?view=uk

If we give an another thought, we may see a *distant* link between the two senses. The meaning of 'case' in case of printing press refers to the notion of classification. The 'case' in the first example is an

> an instance of a particular situation or set of circumstances

In fact the 'case' used in the domain of law, medicine, and even in Grammar, fall under same sense. All these may be considered as an extension of the notion of classification. This then makes *māmalā* a special meaning derived from *ḍibbā viśeṣa*. And hence the modified sūtra may be expressed as:
māmalā[← ḍibbā viśeṣa]

Appendix G

Stanford Parser Outputs

- He is a doctor

```
dep(doctor-7, width-1)
dep(100-3, =-2)
ccomp(width-1, 100-3)
nsubj(doctor-7, He-4)
cop(doctor-7, is-5)
det(doctor-7, a-6)
```

- I saw the man you love.

```
nsubj(saw-2, I-1)
det(man-4, the-3)
dobj(saw-2, man-4)
nsubj(love-6, you-5)
rcmod(man-4, love-6)
```

- The snake the mongoose attacked hissed loudly.

```
det(snake-2, The-1)
nsubj(attacked-5, snake-2)
det(mongoose-4, the-3)
dep(snake-2, mongoose-4)
dep(loudly-7, hissed-6)
advmod(attacked-5, loudly-7)
```

- I need a cup of tea.

```
nsubj(need-2, I-1)
det(cup-4, a-3)
dobj(need-2, cup-4)
prep_of(cup-4, tea-6)
```

- The growth of software industry in recent years is unbelievable.

```
det(growth-2, The-1)
nsubj(unbelievable-10, growth-2)
nn(industry-5, software-4)
prep_of(growth-2, industry-5)
amod(years-8, recent-7)
prep_in(industry-5, years-8)
cop(unbelievable-10, is-9)
```

- the door was opened by him.

```
det(door-2, The-1)
nsubjpass(opened-4, door-2)
auxpass(opened-4, was-3)
agent(opened-4, him-6)
```

- The door was opened by this key.

```
det(door-2, The-1)
nsubjpass(opened-4, door-2)
auxpass(opened-4, was-3)
det(key-7, this-6)
agent(opened-4, key-7)
```

Appendix H

Link parser outputs

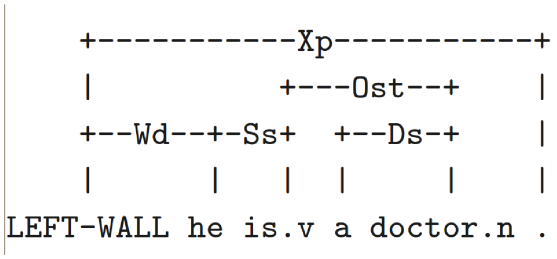


Figure H.1: copula sentence link parser output

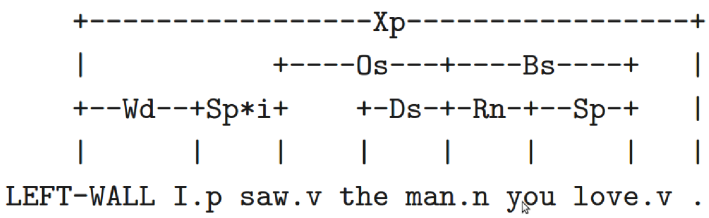


Figure H.2: wh drop sentence link parser output

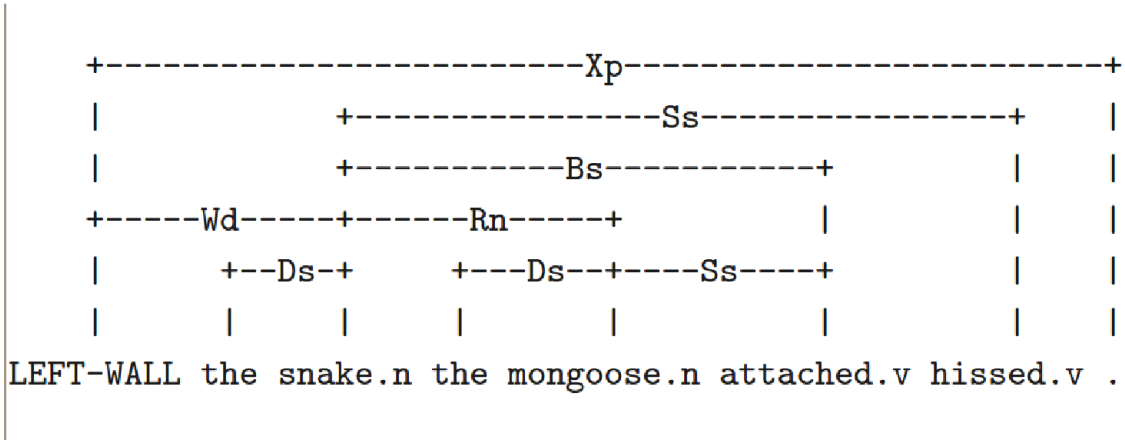


Figure H.3: wh drop sentence link parser output

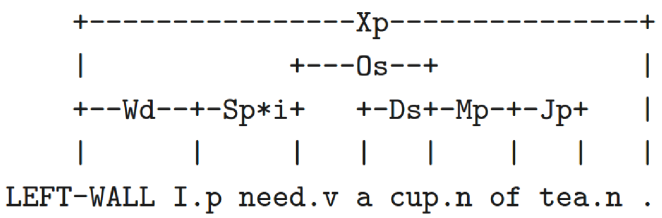


Figure H.4: semantic head example1

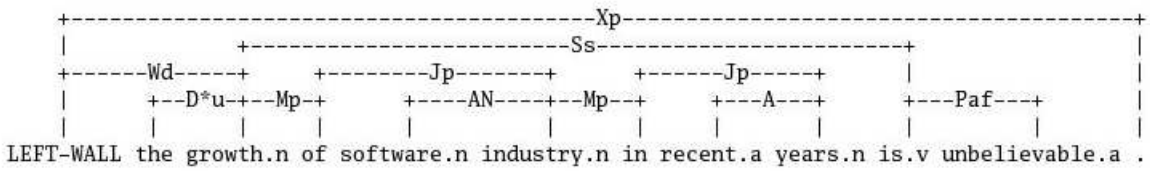


Figure H.5: semantic head example2

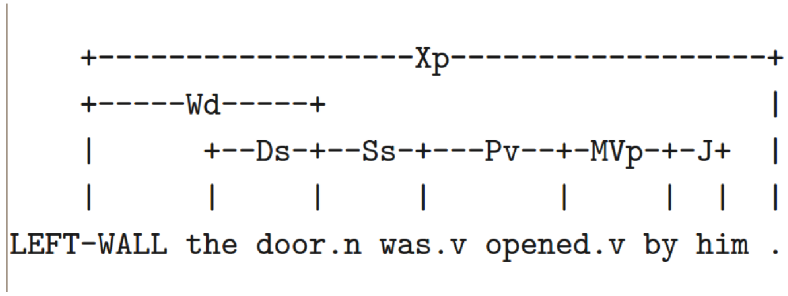


Figure H.6: passive voice example1

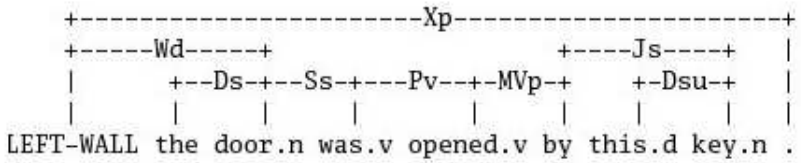


Figure H.7: passive voice example2

Appendix I

Pāṇinian Interface outputs

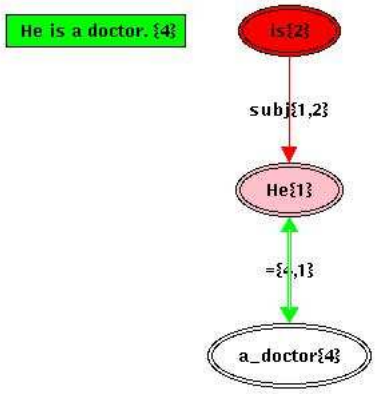


Figure I.1: samaanaadhikarana: PG

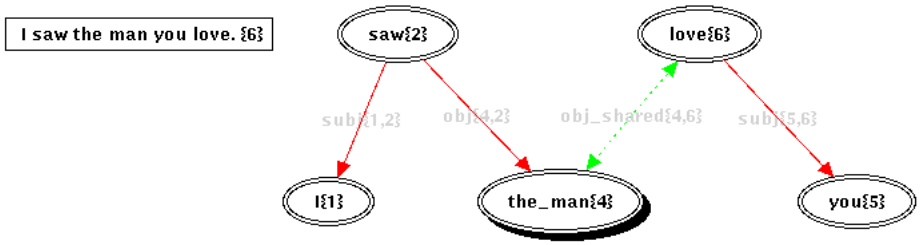


Figure I.2: relative clause: example1;PG

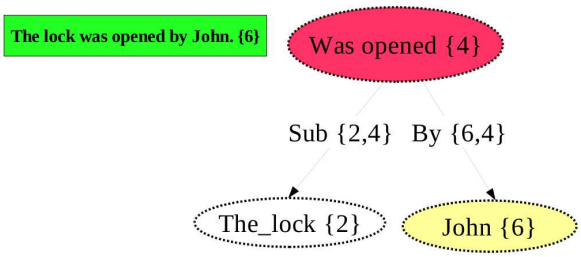


Figure I.3: agent:example1; PG

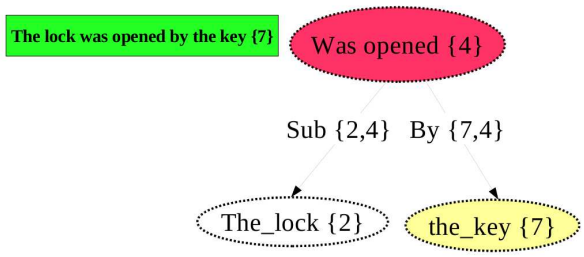


Figure I.4: agent:example2; PG

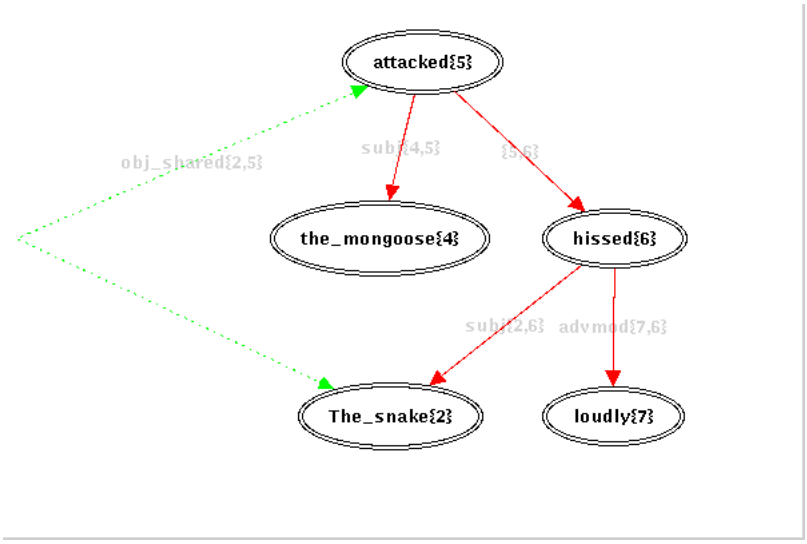


Figure I.5: relative clause: example2;PG

Appendix J

IAST Map table

a	ā	i	ī	u	ū	ṛ	RR
अ	आ	इ	ई	उ	ऊ	ऋ	ॠ
ḷ	e	ai	o	au	ṃ	ṁ	ḥ
ऌ	ए	ऐ	ओ	औ	ं	ँ	ः
		k	kh	g	gh	ṅ	
		क्	ख्	ग्	घ्	ण्	
		c	ch	j	jh	ñ	
		च्	छ्	ज्	झ्	ञ्	
		ṭ	ṭh	ḍ	ḍh	ṇ	
		ट्	ठ्	ड्	ढ्	ण्	
		t	th	d	dh	n	
		त्	थ्	द्	ध्	न्	
		p	ph	b	bh	m	
		प्	फ्	ब्	भ्	म्	
		y	r	l	v		
		य्	र्	ल्	व्		
		ś	ṣ	s	h		
		श्	ष्	स्	ह्		

Appendix K

Extended-Devanagari

The following two charts show the new consonant and vowel symbols added to the Devanāgarī script to represent the vowels and consonants in other Indian scripts faithfully through Devanāgarī.

Figure K.1: Extended-devanagari alphabet (courtesy: ISI Bulletin)

IS 13194:1991 (Refer to the Note on covering page)

Nukta Consonants

क़	ख़	ग़	ज़	ड़	ढ़	फ़
qa	kh̲a	gh̲a	za	d̂a	d̂ha	fa

VOWEL MODIFIERS

Chandrabindu	◌ँ	m̃
Visarg	◌ः	ḥ
Anuswar		◌ं
	before क varg	ṅa
	before च varg	ña
	before ट varg	ṇa
	before त varg	na
	before प varg	ma
	otherwise	ma

Notes:

ऋ, ऌ, ॡ are used only in Sanskrit.
ऍ = short ए in Southern scripts
ऑ = short ओ in Southern scripts
ऍ = new vowel in Devanagari, as in "bat"
ऑ = new vowel in Devanagari, as in "ball"
ऩ n = ன in Tamil
य़ ẏa = य in Bengali and Oriya, while य y = য.
ऱ r̲a = Tamil (ற) Telugu (ఱ), & Malayalam (ഺ)
ऱ r̲a = ர in Marathi
ळ l̤a = used in Marathi
ळ l̤a = Tamil (ள), Malayalam (ള), Telugu (ళ) & Kannada (ಳ)
ऴ z̲a = Tamil (ழ), Malayalam (ഴ)

Figure K.2: Extended-devanagari alphabet (courtesy: ISI Bulletin)

Appendix L

Urdu-Alphabet

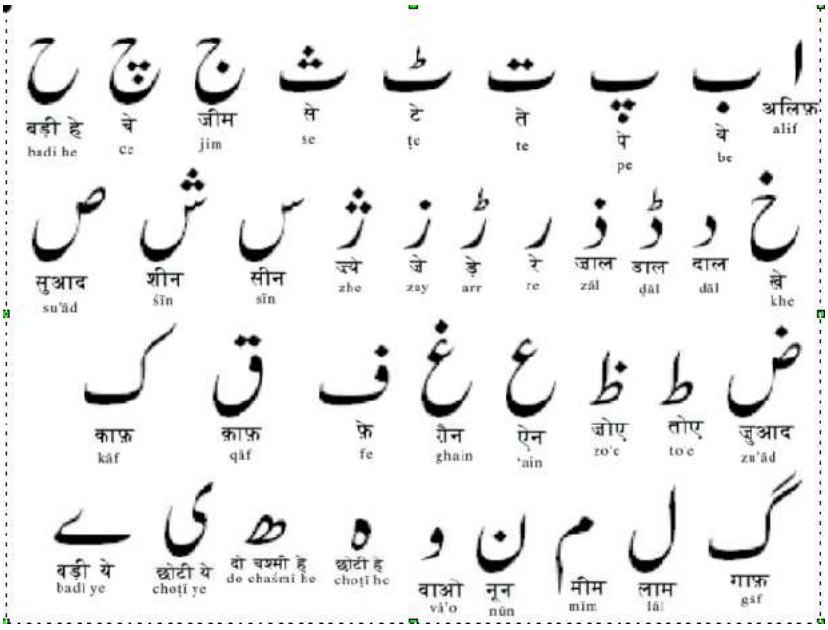


Figure L.1: Urdu Alphabet: Courtesy: en.wikipedia.org/wiki/Urdu_alphabet

Bibliography

[1] Abhyankar and Limaye: *Mahābhāṣya - Dīpikā of Bhartṛhari* Bhandarkar Oriental Research Institute, 1967.

[2] ALPAC *Languages and machines: computers in translation and linguistics.* A report by the Automatic Language Processing Advisory Committee, National Research Council. Washington, D.C. National Academy of Sciences.

[3] Ananthakrishnan R., Hegde J., Bhattacharyya P. and Sasikumar M.: Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation, International Joint Conference on NLP (IJCNLP08), Hyderabad, India, Jan, 2008.

[4] Ananthakrishnan R., P. Bhattacharyya, Sasikumar M. and Shah R. M.: Some Issues in Automatic Evaluation of English-Hindi MT: More Blues for BLEU, ICON 2007, Hyderabad, India, Jan, 2007.

[5] Arnold D. J.: An Introduction to Machine Translation. Oxford: Blackwell, 1994.

[6] Berger A. L., Brown P. F.,Pietra V. J. D., Gillett J. R., Lafferty J. D., Mercer R. L., Printz H., Ures L.: *The Candide System for Machine Translation.* In 1994 Proceedings of the ARPA Conference on Human Language Technology,NJ, USA.

[7] Bharati A. and Sangal R.: *A Karaka Based Approach to Parsing of Indian Languages.*, In *COLING90: Proc. of Int. Conf. on Computational Linguistics (Vol. 3), Helsinki*, Association for Computational Linguistics, NY, August 1990, pp. 25-29.

[8] Bharati A., Sangal R., and Chaitanya V.: *Natural Language Processing, Complexity Theory and Logic.*, In *Foundations of Software Technology and Theoretical Computer Science 10, Lecture Notes in Computer Science 472*, Springer Verlag Berlin, 1990a, pp.410-420.

[9] Bharati A., Chaitanya V., Sangal R.: *Natural Language Processing: A Paninian Perspective..* Prentice Hall of India, New Delhi, 1995.

[10] Bharati A., Bhatia M., Chaitanya V., Sangal R.: *Paninian Grammar Framework Applied to English.* South Asian Language Review, Creative Books, New Delhi, 1997.

[11] Bharati A., Kulkarni A., Chaitanya V., Sangal R., Rao G. U.: *Anusaaraka: Overcoming the Language Barrier in India.* In Anuvaad, Sage Publishers, New Delhi, 2000.

[12] Bharati A., Kulkarni A., Chaitanya V., Sangal R.: *Language Access: An Information Based Approach.* Knowledge-Based Computer Systems, Tata McGraw-Hill, New Delhi, Dec. 2000.

[13] Bharati A., Chaitanya V., Sharma D. M., Kulkarni A.: *Modern Technology for Language Access: An aid to read English in Indian context..* Osmania Papers in Lingustics, vol 26-27, pp:111-126, 2000-01.

[14] Bharati A. and Kulkarni A.: *Machine translation activities in India: A survey.* In proceedings of workshop on survey on Research and Development of Machine Translation in Asian Countries, Thailand, May 13-14, 2002.

[15] Bharati A. and Kulkarni A.: *Design and Architecture of anusAraka: An Approach to Machine.* Translation Satyam Techical Review vol 3, Oct 2003.

[16] Bharati A. and Kulkarni A.: *English grammar from Hindi speaker's point of view in the light of Paaninian grammar. English Grammar from Paninian Perspective, Ed: S Satyanarayan Murty, R J Ramasree, Srinivasa Varakhedi, Rashtriya Sanskrit Vidyapeetham, Tirupati, 2007.*

[17] Bharati A. and Kulkarni A.: *English from Hindi viewpoint: A Paaninian perspective.* Papers on Language Technology Eds: Panchanan Mohanty and S. Arul Mozi, Dravidian University, Kuppam University,2008. first presented at Platinum Jubilee conference of LSI at HCU, Hyderabad, Dec 6-8, 2005

[18] Bokil H. and Bhattacharyya P. Language Independent Natural Language Generation from Universal Networking Language Second International Symposium on Translation Support Systems, IIT Kanpur, India, March, 2002.

[19] Brahmadatt J.: *Ashtadhyayi (Bhashya) Prathamavrtti, three volumes.*, Ramlal Kapoor Trust Bahalgadh, (Sonepat, Haryana, India), 1979. (In Hindi)

[20] Bhattacharya P.: *Some issues in automatic evaluation of English-Hindi MT: more blues for BLUE.* ICON 2007.

[21] Cardona G.: *Panini: A Survey of Research.* Mouton, Hague-Paris, 1978.

[22] Cardona G.: *Panini: His Work and Its Tradition.* Vol. 1: Background and Introduction, Motilal Banarsidas, Delhi, 1988.

[23] Carroll J., Minnen G., and Briscoe T.: *Corpus annotation for parser evaluation.* In Proceedings of the EACL, 1999.

[24] Charniak E.: *A maximum-entropy-inspired parser.* In Proceedings of NAACL-2000.

[25] Charniak E. and Johnson M.: *Coarse-to-fine n-best parsing and MaxEnt descriminative reranking.* In Proceedings of the 43rd annual meeting of the ACL, pp. 173-180, 2005.

[26] Chomsky N.: Principle and parameters in syntactic theory. In Explanation in linguistics eds: N. Hornstein and D.Lightfoot, London: Longman, pp32-75.

[27] Collins M.: *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania., 1999.

[28] Comrie B.: *Language Universals and Linguistic Typology.* The University of Chicago Press, 1989.

[29] Copestake A. and Flickinger D.: *An open-source grammar development environment and broad-coverage English grammar using HPSG.* In Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000), Athens, Greece.

[30] Dave S., Parikh J., Bhattacharya P.: *Interlingua based English-Hindi Machine Translation and Language Divergence.* Journal of Machine Translation (JMT), Volume 17, September, 2002.

[31] Deshpande M. M.: *The meaning of NOUNS: Semantic Theory in Classical and Medieval India.* D.K.PrintWorld New Delhi, 2007.

[32] Dorr B. J.: Machine translation divergences: A formal description and proposed solution . Computational Linguistics 20(4): 597634,1994.

[33] Dvivedi K.: arthavijñāna aura vyākaraṇadarśana Allahabad, 1951

[34] Farwell D. and Wilks Y.: *Ultra: A Multi-lingual Machine Translator.* New Mexico State University.

[35] Frederking R.,Cohen A., Cousseau P., Grannes D. and Nirenburg S.: The Pangloss Mark I MAT System. Proceedings of EACL-93, Utrecht, The Netherlands, April, 1993.

[36] Guleri C. S.: *Guler kī amara kahāṁniyāṁ.* lipi prakashan, Delhi, 1975.

[37] Goyal P. and Sinha R. M. K.: Translation Divergence in English-Sanskrit-Hindi Language Pairs. A Kulkarni and G Huet(eds): Sanskrit Computational Linguistics, LNCS 5406, pp. 134-143. Springer-Verlag Berlin Heidelberg 2009.

[38] Gopinathan S., and Kandaswamy S.: anuvad ki samasayen [Problems of Translation]. Eds, Lokbharti Prakashan, 1993.

[39] Iida E. S. and Iida H.: *Experiments and prospects of example-based machine translation.* ACL, pages 185–192, 1991.

[40] ISLE: *The ISLE classification of machine translation evaluations.* draft 1. A document by the International Standards for Language Engineering.

[41] Hovy E. H., and Nirenburg S.: Approximating an inter-lingua in a principled way. In Proceedings of the DARPA Speech and Natural Language Workshop, New York: Arden House (1992).

[42] Hovy E. H. and Gerber L.: *MT at the Paragraph Level: Improving English Synthesis in SYSTRAN.* Proceedings of the Conference on Theoretical and Methodological Issues in MT (TMI-97).

[43] Hutchins W. J. and Somers H. L.: *An Introduction to Machine Translation.* London: Academic Press, 1992.

[44] Hutchins J.: *Compendium of Machine Translation Software.* Available from the International Association of Machine Translation (IAMT).

[45] Jhalakikara nyāyakośa: A dictionary of Technical Terms of Indian Philosophy Ed. Abhyankar, Bombay Sanskrit and Prakrit Series, 49, Poona, 1928

[46] Joshi A. K.: *Tree Adjoining Grammar.* In D. Dowty et.al. (eds.) Natural Language Parsing, Cambridge University Press, 1985.

[47] Joshi S. D.: *Patanjali's Vyakarana Mahabhashya, (several volumes).* Univ. of Poona, Pune, 1968.

[48] Joshi S. D. and Roodebergen J. A. F.: *The Aṣṭādhyāyīof Pāṇini.* (several volumes), Sahitya Akademi, Delhi, 1998.

[49] Kay M.: *The proper place of men and machines in translation.* Machine Translation 23., 1997.

[50] King T. H., Crouch R., Riezler S., Dalrymple M., and Kaplan R.: *The PARC 700 dependency bank.* In 4th International Workshop on Linguistically Interpreted Corpora (LINC-03).

[51] King M. and Perschke S.: *Machine Translation Today: The State of the Art.* Edinburgh University Press., 1987.

[52] Knight K., Chander I., Haines M., Hatzivassiloglou V., Hovy E., Iida M., Luk S., Okumura A., Whitney R., Yamada K.: Integrating Knowledge Bases and Statistics in MT. Proc. of the Conference of the Association for Machine Translation in the Americas (AMTA), 1994.

[53] Knight K., Chander I., Haines M., Hatzivassiloglou V. , Hovy E., Iida M., Luk S., Okumura A., Whitney R., Yamada K.: Example-Based Machine Translation in the Pangloss system. Proceedings of the 16th conference on Computational linguistics, Copenhagen, Denmark, 1996.

[54] Kilgarriff A.: An evaluation of a lexicographer's workbench incorporating word sense disambiguation . Proc. CICLING, 3rd Int Conf on Intelligent Text Processing and Computational Linguistics, Mexico City. Springer Verlag, 2003.

[55] Kiparsky P.: *Some Theoretical Problems in Panini's Grammar.*, Bhandarkar Oriental Research Institute, Poona 411004 India, 1982.

[56] Kiparsky P.: *On the Architecture of Panini's Grammar.* CIEFL, Hyderabad, Jan 2002.

[57] LAurel B.: The art of Human Computer Interface Design. Addison Wesley, 1991.

[58] Levin B.: English verb classes and alternations: a primary investigation. Chicago, University of Chicago Press, 1993.

[59] Lin D.: Dependency-Based evaluation of MINIPAR. In workshop on the evaluation of Parsing Systems, Granada, Spain, 1998.

[60] Maṇikaṇa: A Navya nyāya Manual, Adyar library series, 88, Madras, 1960

[61] Klein D. and Manning C. D. : *Accurate unlexicalized parsing.* ACL 2003. pp. 423-430.

[62] Marneffe M., MacCartney B. and Manning C. D.: *Generating Typed Dependency Parses from Phrase Structure Parses.* LREC-06.

[63] Massica C. P.: The Indo-Aryan Languages. Cambridge University Press, 1991.

[64] Maegaard B., and Hansen V.: PaTrans Machine Translation of Patent Texts. From Research to Practical Application. In Convention Digest: Second Language Engineering Conference (pp. 1–8). London, 1995.

[65] Malik A., Boitet C. and Bhattacharyya P.: Hindi-Urdu Transliteration Using Finite State Transducers, Computational Linguistics (COLING08), Manchester, UK, August, 2008.

[66] Mohanty R., Almeida A., Srinivas S. and Bhattacharyya P. The complexity of OF in English International Conference on Natural Language Processing, Hyderabad, India, December 2004.

[67] Mohanty R., Almeida A. and Bhattacharyya P.: Prepositional Phrase Attachment and Interlingua, International Conference on Intelligent Text Processing and Computational Linguistics (CCLING-2005) Workshop on UNL and other Interlingua and their Applications, Mexico City, Mexico, February, 2005.

[68] Murthy K. N.: *Universal Clause Structure Grammar and the Syntax of Relatively Free Word Order Languages.* South Asian Language Review, Vol VII, No 1, Jan 1997, pp 47-64.

[69] Murthy K. N.: *Natural Language Processing: An Information Access Perspective.* New Delhi: Ess Ess Publications., 2006.

[70] Narayana V. N.: *Anusaraka: A Device to Overcome the Language Barrier.* Ph.D. thesis, Dept. of CSE, I.I.T. Kanpur, 1994.

[71] Niemann H., Noeth E., Kiessling A., Kompe R. and Batliner A.: Prosodic Processing and its Use in Verbmobil. Proceedings of ICASSP-97 (7578). Munich, Germany, 1997.

[72] Nirenburg S.: *Machine Translation: Theoretical and Methodological Issues.* Cambridge: Cambridge University Press., 1987.

[73] Nirenburg S., Carbonell J. C., Tomita M., and Goodman K.: Machine Translation: A Knowledge-Based Approach. San Mateo, CA: Kaufmann,1992.

[74] Radford A.: *Syntax: A minimalist Introduction.* Cambridge University Press, 2002.

[75] Radford A.: *English Syntax: An introduction.* Cambridge University Press, 2004.

[76] Raja Kunjunni K.: Indian theories of Meaning, The Adyar library and Research Center, Madras, 1963

[77] Rao D., Bhattacharyya P. and Mamidi R. Natural Language generation for English to Hindi Human Aided Machine Translation International Conference on Knowledge Based Computer Systems (KBCS'98), Mumbai, December, 1998.

[78] Ruhl C.: On Monosemy: A study in Linguistic Semantics Albany: State University of New York Press

[79] Sag I. A. and Thomas W.: *Syntactic Theory: A formal Introduction.* CSLI publications, 1999.

[80] Sapir E.: The status of linguistics as a science. Language 5. 207-14. Reprinted in The selected writings of Edward Sapir in language, culture, and personality, ed. by D. G. Mandelbaum, 160-6. Berkeley: University of California Press.

[81] Sato S., Nagao M.: Toward Memory-based Translation. Proceedings of COLING 1990, Finland.

[82] Saxton K. L.: A monosemic semantics of 'just' Proceedings of the 1994 Annual Conference of the Canadian Linguistic Association Actes du congrs annuel de l'Association canadienne de linguistique 1994 University of Calgary, Calgary, Alberta

[83] Schutz J., Thurmair G.: *An architecture sketch of Eurotra-II.* In MTS 91.

[84] Singh S.: *English - Hindi Translation Grammar.* Prabhat Prakashan, Delhi, 2003.

[85] Sleator D. D. and Temperley D.: Parsing English with a link grammar. In third international Workshop on Parsing Technologies, 1993.

[86] Sleator D. D. and Temperley D.: *Parsing English with a link grammar.* In Third International Workshop on Parsing Technologies,1993.

[87] Srinivas M. and Bhattacharyya P.: Prepositional Phrase Attachment through Semantic Association using Connectionist Approach, 3rd Global Wordnet Conference (GWC 06), Jeju Island, Korea, January, 2006.

[88] Steiner: After Babel: Aspects of Language and Translation. London: Oxford University Press, 1975.

[89] Tognazzini B.: Tog on Interface. Addison Wesley, 1991.

[90] Uchida H.: *ATLAS-II: A machine translation system using conceptual structure as an interlingua .* MTS 89.

[91] Whorf B. L.: Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf. J. B. Carroll, (Ed.) Cambridge, MA: MIT Press, 1956.

[92] Yamron J., Cant J., Demedts A., Dietzel T., Ito Y.: The automatic component of the LINGSTAT machine-aided translation system. Proceedings of the workshop on Human Language Technology, NJ, 1994.

[93] Yarowsky D.: *Three Machine Learning Algorithms for Lexical Ambiguity Resolution.* Ph.D. thesis, University of Pennsylvania, Department of Computer and Information Sciences., 1995.

[94] Yusuke M., and Junichi T.: *Probabilistic Disambiguation Models for Wide-Coverage HPSG Parsing.* In Proceedings of ACL-2005, pp. 83-90.

[95] http://www.iiit.net/ltrc/index.html

[96] http://www.cyc.com

[97] http://www.ncst.ernet.in/matra

[98] http://www.tdil.gov.in

[99] http://www.undl.org

[100] http://www.mt-archive.info/methods-2.htm

[101] http://www.isi.edu/natural-language/mteval

[102] http://www.hf.ntnu.no/engelsk/staff/johannesson/111gram/lect17.htm

[103] CMU's Link parser: ftp://ftp.cs.cmu.edu/user/sleator/link-grammar

[104] General Public License: http://www.gnu.org/copyleft/gpl.txt

[105] Minipar: http://www.cs.umanitoba.ca/lindek/minipar.htm