Chapter 2

Literature Survey

“I have a lack of fear, whereas in the past the fear of failure was a powerful motivator. Anyway, I have great expectations for the future, but I just don't know if I'm the monarch of all, I survey.”

-Sylvester Stallone

This chapter presents panoramic view of MT history and development through literature survey. Literature survey helps the researcher to understand the history, past efforts, conventional approaches and present state-of-the-art about his/her research problem. This understanding further helps in deciding due course of action towards solving the research problem understudy. For our research problem, we have done extensive literature survey of MT field in terms of various approaches/architectures and MT development paradigms along with survey of various already developed MT systems around the globe as well as in Indian context. Since our proposed MT framework is rule based, we have also surveyed various Grammar formalisms under constituency and dependency schools.

2.1 MT History

The history of MT is divided in five time slabs. Pre-computer era, era of optimism or golden era (1945-1965), Post ALPAC era (1966-1980) (Hutchins J. , 14 June 1996), (ALPAC, 1966), Renaissance period and 90’s to modern Internet era (Hutchins J. W., 1986), (Dorr B. J., 1999). The history of MT dates back to the 17th century i.e. pre computer era. In 1629, René Descartes, a French philosopher, proposed a universal language, based on the idea of sharing one symbol in different tongues (Hutchins J. ). Real activities started in the mid 1930s, when French-Armenian Georges Artsrouni and a Russian Peter Troyanskii applied for patents for ‘translating machines’. Georges Artsrouni had designed a storage device using paper tape which was capable of finding the equivalent words of given word in Source Language(SL), in other
languages. He demonstrated prototype of his system 1937. The Russian, Peter Smirnov-Troyanskii, proposed more significant idea where he envisaged three stages of mechanical translation based on Analysis-Transfer-Generation scheme, in which two human language editors (each knowing either SL or TL) were required along with one machine. According to his idea an editor knowing only the source language would carry out the ‘logical’ analysis of sentence words to their base form and syntactic functions and subsequently machine would transform these sequences of base forms and functions into equivalent target language sentences using certain abstract/symbolic representation, this would then be converted by another editor knowing only the target language into the normal forms of target language. However his patent referred only to the mechanical machine which would undertake the second stage, Troyanskii strongly believed that “the process of logical analysis could itself be mechanised”. Unfortunately Troyanskii was not well known person, outside Russia and he had proposed his idea quite before the right time, where technology hadn’t advanced to the stage which could support his idea (Somers, 1992). After evolution of computers, similar idea was further taken up, by Warren Weaver of the Rockefeller Foundation, when Weaver wrote a letter in March 1947 to cyberneticist Norbert Wiener seeking his opinion on the idea. Paragraph from his letter says:

“I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.”

(Weaver 1947)

Weaver also discussed this idea with Andrew D. Booth, a British crystallographer, who further explored it but it was in 1949, it was brought to general notice, Weaver wrote a Memorandum on Translation which proved to be influential for driving the MT further (Somers, 1992). It was followed by first conference on MT in 1952. MT activity was formally triggered and public demonstration of first ever MT in the history developed at George Town University in collaboration with IBM was made in New York. This MT system was built for Russian to English pair. It was consisting of 40 sentences. This exercise is known as ‘George Town experiment’ in the history of MT. It gave boost to MT research in USA and Europe. The further period called era of
optimism, when lot of MT work started, huge investments were made. In 1959, IBM installed an MT system for the United States Air Force, followed by Georgetown University installing systems at Erratum and the United States Atomic Energy Agency. As things were moving ahead, and researchers started handling complex text, delusions started growing and situation worsened, when semantic barriers started coming in to picture. Meanwhile output of MT systems like Mark-II MT system (developed by IBM and Washington University) installed at the USAF Foreign Technology Division, and the Georgetown University system, which was installed at the US Atomic Energy Authority, was found to be not satisfactory. It made investors to press the panic button and in 1964 wave of panicky was observed in sponsors and investors. Immediately after that an expert committee Automatic Language Processing Advisory committee (ALPAC) was formed under the chairmanship of John R. Pierce (Bell Telephone Laboratories). Other members of the committee included John B. Carroll (Harvard University), Eric P. Hamp (University of Chicago), David G. Hays (RAND Corporation), Charles F. Hockett (Cornell University), Anthony G. Oettinger (Harvard University), and Alan Perlis (Carnegie Institute of Technology). In this committee, Hays and Oettinger were only members, who were active MT researchers. The major objective to setup ALPAC was to examine the prospects of MT research and output. In its influential 1966 report (Hutchins J., 14 June 1996), (ALPAC, 1966), ALPAC concluded that MT was slower, less accurate and twice as expensive as human translation and further stated that “there is no immediate or predictable prospect of useful Machine Translation”. Hence there was no need for further investment in MT research; instead it recommended the development of machine aids for translators, such as automatic dictionaries, and recommended support of basic research in computational linguistics. The ALPAC report was widely condemned as narrow, biased and short sighted, it was certainly wrong to criticize MT because output had to be post-edited, and it misjudged the economic factors due to which large-scale financial support for then prevailing approaches could not continue. Its influence was so profound that it brought a virtual end to MT research in the United States for over a decade and damaging the public perception of MT for many years afterwards. Though ALPAC report was restricted only to United States, but it spread a wave of disappointment and depression amongst the MT researchers worldwide and brought MT activities to standstill in USA, Europe and Soviet Union. In spite of these
facts, ALPAC report also had couple of good things like promotion of further research in Computational Linguistics & MT evaluation procedures. The study by John B. Carroll regarding evaluation of both human and machine translations, as part of ALPAC evaluation exercise had great influence on many MT evaluation techniques for subsequent years, (Hutchins J., 14 June 1996) (ALPAC, 1966). Research continued in Canada, France, and Germany. Ray of hope was seen, when working MT system ‘SYSTRAN’ was installed at USAF (1970) followed by the Commission of the European Communities (1976). In the same year METEO system (1976) for translating weather reports, appeared on the MT canvas. METEO was developed at Montreal University, Canada. Late 70’s period was special as there was growing need for multilingual MT systems, for commercial and technical documentation purposes. Year 1980 observed signs of renaissance as several types MT systems from wide number of countries emerged. During this era, due to advances in Computer Hardware, more powerful still low cost machines were available at disposal for MT researchers. New approaches and techniques were invented by MT scholars, the dominant approach like Interlingua i.e. indirect translation using intermediary representation was result of such initiatives. Good number of research projects came up during this period, some of the notable projects included: GETA-Ariane (Grenoble), SUSY (Saarbrucken), Mu (Kyoto), DLT (Utrecht), Rosetta (Eindhoven), knowledge-based project at Carnegie-Mellon University (Pittsburgh), two international multilingual projects: EUROTRA, CICC project with participants in China, Indonesia and Thailand. SYSTRAN was expanded for other language pairs, other MT systems like Logos (English-German, English-French), PAN American Health Organization’s MT system (English-Spanish and two ways), METAL (German to English) system and English-Japanese Bi-directional MT system by companies in Japan also came up. The early 90’s was a major turning point as three milestones were reached in MT history. Firstly, emergence of non-linguistic based paradigms for MT, speech came within the ambit of MT and change in the focus from pure research to practical MT applications. IBM research group lead by Peter Brown in 1990, (Brown P., Della-Pietra, Jelinek, Mercer, & Roossin, 1988), (Brown P. F., 1990) published the results of their experiments on French to English MT system “Candide”, which were purely based on statistical methods (Brown, et al., March 1994). Secondly, Japanese scholars M. Nagao proposed a new approach for
translation based on parallel corpora of translation examples (Nagao M., 1984), this approach is now known as ‘example-based’ machine translation (EBMT). Both approaches were bold in the sense that their distinctive feature was that they did not use any syntactic or semantic rules and still they had scientific base and were not purely ad-hoc approaches. This was a new paradigm shift in history of MT. On speech translation side a good number initiatives were started in Japan, USA and Germany. Speech translation involved integration of speech recognition, speech synthesis systems with MT modules. The speech translation projects included ATR (Nara, Japan), JANUS project (ATR, Carnegie-Mellon University and the University of Karlsruhe), Verbmobil (Germany) projects (Emele, 1996).

Advent of Internet and huge penetration of PC machines further widened the scope and demand of MT software and MT went further high tech in the sense that MT was made available as online service by providers like Systran, BabelFish, Altavista. Internet imposed new constraints of faster response (time) and fully automatic MT systems, on MT research. From research point of view SMT and EBMT approaches were predicted to rule for coming decade of new millennium. After 2000, search engine giant Google entered the MT arena with their online MT engine Google Translate (2007), presently Google Translate has 66 language pairs. Microsoft has also come up with their Bing Translator engine.

The advent of SMT and OSI (Open Source Initiative) has taken MT research from Ivory towers to common place so that various tools as well as complete MT suites (Moses, Logos, Apertium, and GIZA) are available at the disposal of any researcher in any corner of the world. Tool development is otherwise biggest bottleneck in MT research, which needs long time and big efforts. Researchers can now better contribute by concentrating on linguistic aspects. Global as well as Indian MT initiatives are discussed in detail in the following section of this chapter.

2.2 MT Paradigms

While the MT pyramid (Pl. see Figure 1.3) discussed in the previous chapter describes the processing architecture i.e. actual processing design (direct, transfer, Interlingua), MT paradigm describes informational components, which aids the processing design (architecture). MT architecture and paradigms are two axes of comparison in MT
space. Different paradigms with internal process variation can be built using same architecture and one paradigm can be used to build systems based on different architectures with varying level of analysis. Hence there is overlapping of paradigms & architectures, due to which there are differences amongst the MT researchers regarding MT paradigms as the boundaries between them are very thin. Figure 2.1 below shows complete taxonomy of approaches, architectures and paradigms. This is new dimension in the MT research.

This section discusses some of the MT paradigms that have evolved due to painstaking efforts by research scholars over the period of time. The list has been prepared from various important conferences like Association of Computational Linguistics (ACL), TMI conferences (International conference on Theoretical and Methodical issues in MT), COLING (conference on Computational Linguistics) and MT summits etc. MT paradigms are broadly classified as those that heavily rely on linguistic knowledge/technique and MT systems that do not use any linguistic knowledge at all. Further, we can also have MT systems that combine earlier two (hybrid).

![Figure 2.1 MT Approaches, Architectures and Paradigms: Aggregate View](image)

**2.2.1 Linguistic knowledge based Paradigms**

This class of MT paradigms include Rule based MT (RBMT) (Arnold, Kruwer, Tombe, & Sadler, 1988), (Arnold & Sadler, September, 1990), (Landsbergen, Odijk,
& Schenk, 1989), (Kaplan, Netter, Zaenen, & Wedekind, 1989), Knowledge based MT (KBMT) (Carbonell, 1987), (Lavie, 2005), (Nirenburg S., 1992), Lexicon based MT (LBMT) (Abeille, Schabes, & Joshi, 1990), Constraint based MT (CBMT) (Eberle, Kasper, & Rohrer, 1992), (Arnold & Sadler, August, 1992), (Kaplan & Wedekind, 1993), Shake & Bake MT (SBMT) (Whitelock, 1991), (Whitelock, 23-28 August 1992), (Beaven, 1992), (Brew, 1992), Principle based MT (PBMT) (Dorr B. J., 1991), (Dorr B. J., 1993). These paradigms use some kind of knowledge base for their processing, which includes grammatical features, semantic features, grammar rules, etc. we have extracted highlights of each of them from the literature (Dorr B. J., 1999), (Arnold, Balkan, Meijer, Humphreys, & Sadler, 1994). The same are presented in the following sections (2.3.1) and (2.3.2).

2.2.1.1 Rule Based MT (RBMT)

RBMT is basic and popular scientific approach for building practical MT systems. It can be used with Transfer and Interlingua architectures as well as data driven approach. Different paradigms have evolved from RBMT, hence they seem to be overlapping with RBMT. RBMT systems rely on set of rules at different linguistic abstraction levels of translation process. Good discussion is found on this approach in literature (Appelo., 1986), (Tombe & Arnold, 1987), (Samvelian, Danlos, & P.Samvelian, 1992), (Fujii, Suzuki, Maruyama, & Dasai, 1990), (McCord, 1989), (Landsbergen, Odijk, & Schenk, 1989). MT systems like MiMo (Arnold & Sadler, September, 1990), (Noord, Dorrepaal, Eijk, Florenza, & Tombe, 1990), METAL (Thurmair, 1990), EUROTRA (Tombe & Arnold, 1987), Rosetta prototype Interlingua system uses RBMT (Rosetta, 1994) (Rosetta, 2013). Rosetta uses ‘M’ (meaning preserving) rules and ‘S’ (non-meaningful rules) rules for the phenomena of composition and exception respectively. Rule ranking and exception handling are important issues from development point of view for RBMT based systems.

2.2.1.2 Knowledge Based MT (KBMT)

This is a popular and ranked as most sophisticated paradigm under this category (Lavie, 2005), (Nirenburg S., 1992), (Carbonell, 1987), (Nyberg, Mitamura, G., & Carbonell, July 1991), (Nirenburg B. O., 1995). KBMT concentrates on developing knowledge intensive morphological, syntactical, semantic information for lexicons.
Most of the systems built using this paradigm use transfer as well as Interlingua architecture. It follows the classical steps of processing i.e. analysis and generation using rich knowledge base. MT systems based on this approach use rich Lexicons, SL/TL as well as Transfer Grammar. Grammar rules for POS tagging/Morphological analysis. KBMT based MT systems include ULTRA (Farwell & Wilks, July 1991), Pangloss (Carnegie Milan University), Japangloss and GAZZELLE at University of Southern California (K. Knight, August 1995), (Luk., 1994), UNICON at university of Pennsylvania (Wu, 1995).

2.2.1.3 Lexicon based MT (LBMT)

Any system that supplies rules for mapping lexical entries of one language to another is said to be based on LBMT paradigm. It is said to overlap heavily with RBMT and S&BMT approaches. This paradigm has been studied many scholars (Blaaser, Schwall, & Storrer, 1992), (Dorr, 1992), (Dorr, 1993), (Farwell & Wilks, July 1991), (Trujillo, 1992), (Fujita & Kimikazu, 1991). Most of the popular MT systems like C-DAC’s MANTRA (AAI Group; C-DAC, Pune, 1997) etc. based on this approach. These systems use Tree Adjoining Grammar (TAG) (English-French, French-English, and English-Indian Languages), (Abeille, Schabes, & Joshi, 1990). Maximum information is stored with lexical items. TAG based systems can handle word order divergences very elegantly. This paradigm can neatly handle categorical divergence. Only downside is that it requires the tree structures (LTAG trees) to be stored in the lexicon.

2.2.1.4 Constraint based MT (CBMT)

MT systems based on this approach make use of constraints on combination of lexical items or constituents depending upon and architecture and level of analysis. The constraint in case of lexical level may be presented using feature structures (F-structure) in LFG (Lexical functional grammar) like grammar formalism. The LFG based MT Systems (Eberle, Kasper, & Rohrer, 1992), (Kaplan, Netter, Zaenen, & Wedekind, 1989), (Kaplan & Wedekind, 1993), (Arnold & Sadler, August, 1992), (Whitelock, 23-28 August 1992) are example of this type. This paradigm helps to handle Head-Switching divergences because the syntactic and f-structure are tightly coupled with some mediating selectional constraints associated with lexical entries.
This tight coupling is sometimes disadvantage as some grammatical concept might be represented using more than one ways. Also it is reported that it can't handle the cases of embedded clauses (L. Sadler, 1991).

2.2.1.5 Principle based MT (PBMT)

This paradigm also has been tried by different scholars (Dorr B. J., 1991), (Dorr B. J., 1993), (Dorr B. J., 1992), (Sharp, 1985), (M.Eeg-Olofsson & B.Sigurd, 2009), (Wehrli, 1992). This paradigm is treated as alternative to RBMT in which rules are replaced by set of principles that cater for morphological, grammatical, and lexical phenomena like inflectional and derivational morphology, passivation, transformation, negative sentence formation and causitivization (in Indian languages) etc. Parsing in PBMT is based on theory of Principles and parameters described in Chomsky’s Generative grammar (Chomsky N., Feb 1957). Grammar is thought of as set of language independent principles and set of language dependent parameters used for parsing particular language. PRINCITRON: Korean to English is an example of PBMT (Dorr, Lin, Lee, Suh, & Sungki, 1995). In this approach structure building is deferred until underlined descriptions satisfy set of principles. Due to its anatomy, it can be used to build MT system on Interlingua architecture. PBMT Processing mechanism is language independent and accommodates many structurally different languages. PBMT can successfully handle the thematic divergence (M.Eeg-Olofsson & B.Sigurd, 2009). PBMT is complimentary to EBMT and KBMT.

2.2.1.6 Shake & Bake MT (S&B MT)

This is yet another interesting linguistic based approach. The proponents of this approach claim this approach to be an effective alternative to transfer approach (Whitelock, 1991), (Whitelock, 23–28 August 1992), (Beaven, 1992), (Brew, 1992). In this paradigm, translation takes place using Shake and Bake phases. Shake refers to SL analysis (parsing), whereas Bake refers to generating the TL by combining the TL words using TL composition rules governed by ‘bilingual SL-TL’ mappings. In simple words this approach parses the SL and then SL words are transferred to TL side where TL sentence construction is done based on TL grammar so that major issues of word order, divergences due to non-availability of mapping rules for complex compositional rules can be elegantly handled. As mentioned earlier, transfer
rules are defined on the basis of ‘bilingual mappings’ of lexical items from SL to TL. After the SL sentence is parsed, TL words are collected from correctly parsed SL words; the target sentence is generated by combining the TL lexical items obtained from ‘bilingual mappings’. This approach can handle Head switching divergence in complex translations as well as GNP, Categorical divergences along with word order reordering issues. Only disadvantage of this approach as described in (Brew, 1992) is that the problem of reordering TL words is computationally ‘NP complete’.

2.2.2 Non-Linguistic Paradigms

These paradigms fall under data driven approach as they are based on large corpora. Main contenders in this approach are Statistical MT (SMT) (Brown P. F., 1990), Example based MT (EBMT) (Nagao M., 1984), Dialogue based MT (DBMT) (Boitet, 1989) and Neural Network based MT (NNMT) (Scott, 1990), (Higinbotham, 1990). Due to advent of Internet large corpora is available to NLP community, hence statistical methods have become the latest trend in MT and NLP research. DBMT is an interactive system in which translation is done based on interaction between user and system, while NNMT attempts to exploit Neural Networks for MT purpose. In nutshell paradigms under this category exploit statistical and Machine Learning techniques.

2.2.2.1 Statistical MT (SMT)

This paradigm was proposed by Peter F. Brown (Brown P. F., 1990), (Brown P. F., Della-Pietra, Jelinek, Mercer, & Roossin, 1988), at IBM research center, NY (USA). This paradigm is based on purely statistical techniques. In layman’s words this approach can be described as translating a SL sentence to TL sentence by translating each word of SL sentence to possible TL word, which has highest probability of being the exact TL equivalent. This probability between SL & TL word pairs is computed \textit{a priori} from the bilingual lexicons. So to say probabilities of all combinations of SL and proposed TL words (options) are added and only SL-TL word combination having maximum probability (maximum sum) is chosen as TL translation of SL sentence. In statistical terms, this problem is specified using following equation:

\[
\Pr(S,T) = \arg\max_S \Pr(S) \cdot \Pr(T/S)
\]
Where,

$S$: Source Language Sentence

$T$: Target Language Translation

$Pr(S, T)$: Joint probability that $S$ & $T$ in which $T$ is translation of $S$

$Pr(S)$: Source Language Model

$Pr(T/S)$: Translation Model

This technique needs huge bilingual corpora. The system based on this approach has to undergo training phase a number of times. After every training iteration, training phase refines the translation model. The word alignments can also be computed statistically. Various tools like GIZA++ are available for trying SMT (Google), (Koehn, 2010). The complete SMT suite MOSES is also available for SMT research (Statmt/Moses). This approach can be extended from computing single word probabilities to phrases of ‘$n$’ words to gather called ‘$n$-gram’ based translation. Higher the value of ‘$n$’ better is the quality of translation. This approach is being tried by many MT researchers across the globe for different language pairs.

2.2.2.2 Example based MT (EBMT)

This paradigm was first proposed M. Nagao (Nagao M. , 1984), it is based on simple concept of ‘translation by analogy’. It assumes huge parallel aligned corpora. Whenever system encounters a sentence, it first searches for it in the corpora, if it is present, its aligned TL sentence is presented to the user. If the sentence is not present in the database, it is chopped into phrases, which are searched for matching. These systems can really yield high quality MT output, but are subject to availability of huge parallel aligned corpus. Along with phrase matching, sentence structure matching is also used in some cases by EBMT systems. Finding complete input sentence in Database is the best case, which is not so frequent phenomena hence most EBMT systems are restricted to sub problems like function word, noun phrase, verb phrase and prepositional phrase matching. Restricted form of commercially available EBMT is known as ‘Translation Memory’, in which, newly translated sentences are stored in the database so that next time whenever the user inputs the new version of same sentence, the stored translation can be presented to the user. This concept is becoming
popular and used in Translation Bench softwares. In other forms of translation memory, inexact but close forms of translation are also presented because of the assumption that, time required to edit close translation is less than from scratch translation.

2.2.2.3 Neural Network based MT (NNMT)

This is a recent and new idea. Neural Network (NN) technology can be used in performing important functions in MT process like parsing (Jain, 1991), lexical disambiguation (Higinbotham, 1990), and grammar rule learning. Handling complete MT process may require complex representation and large amount of time for training NN hence it is not feasible as on today. But Mclean (McLean, 1992) showed that NN can be used to match sentences/phrases in an EBMT system. So, instead of designing core MT philosophy with NN, subtasks of MT process can be performed with the help of NN.

2.3 MT Initiatives

Historically, MT initiative started with Russian-English pair but due to further research in the MT field, these initiatives were extended to other language pairs also. Good amount of work has been done for European Language pairs with reasonably good quality results. The MT efforts can be primarily classified as MT systems catering the sister language pairs and non-sister language pair. The need for cross language family MT systems aroused due to various factors such as warfare, business and diplomatic reasons etc. In the course of our study, we have surveyed some of classical MT systems as well as MT systems in Indian Context.

2.3.1 Global Context

Though, we have mentioned most of the important western MT systems in the History section 2.1) above, various historical MT systems as extracted from various sources (Arnold, Balkan, Meijer, Humphreys, & Sadler, 1994), (Goyal, 2010), (Hutchins J. ), were surveyed to understand their approach and features, these systems are summarized in the following table (Table 2.1).
## Table 2.1 Historical MT Systems

<table>
<thead>
<tr>
<th>Year</th>
<th>MT System</th>
<th>Language Pair</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1952</td>
<td>GAT (Georgetown University &amp; IBM) (Hutchins W. J., 2004), (Hutchins J. , 2005) (Dostert, 1955)</td>
<td>Russian-English</td>
<td>Direct Translation</td>
</tr>
<tr>
<td>1961</td>
<td>CETA Grenoble University, France (Vauquois &amp; Boitet, 1985), (Veillon, 1968), (Vauquois B. , 1976),</td>
<td>Russian-French</td>
<td>Interlingua and Dependency Structure Analysis. In 1971 improved system GETA was developed based on CETA.</td>
</tr>
<tr>
<td>1961</td>
<td>METAL, Linguistic Research Center, University of Texas, Austin, USA (Thurmair, 1990)</td>
<td>German-English</td>
<td>Indirect Translation using, Chomsky’s paradigm of Analysis-Transfer-Synthesis.</td>
</tr>
<tr>
<td>1964</td>
<td>Mark-II, by IBM research Center, installed at US Air Force’s Foreign Technology Division at the Wright-Patterson Air Force Base, Dayton, Ohio (Orr &amp; Small, 1967)</td>
<td>Russian-English</td>
<td>Direct Translation</td>
</tr>
<tr>
<td>1964</td>
<td>LOG*OS, Logos Corporation, USA (Yurtseven, 29 October-1 November 1997), (Yurtseven, 31 March – 3 April 1997), (Yurtseven, 29 October – 1 November 1997)</td>
<td>Vietnamese-English, English-French, English-German</td>
<td>Morphology, Syntactic and heavy Semantic analysis</td>
</tr>
<tr>
<td>1965</td>
<td>TAUM-AVIATION University of Montreal, Quebec, Canada (Isabelle &amp; Bourbeau, 1985), (Chandioux, 3-6 May 1977)</td>
<td>English-French Weather Forecast domain</td>
<td>Transfer</td>
</tr>
<tr>
<td>1968</td>
<td>SYSTRAN Developed by Hutchins and Somer (Ryan, September 17-19, 1987), (Senez, 10-13 July, 1995), (Senellart, Yang, &amp; Rebollo, 23-27 September 2003), (Dugast, Senellart, &amp; Kohen, 23 June 2007), (Senellart &amp; Jean, 2005)</td>
<td>Initially English-Russian now works for 29 language pairs</td>
<td>Direct Machine Translation Also provides online translation service</td>
</tr>
<tr>
<td>1968</td>
<td>CULT (Chinese University Language Translator) Chinese University, Hong Kong (Loh, Kong, &amp; Hung, 1978), (P.H.Nancarrow, 1978), (L.Kong &amp; S.C.Loh, 3-6 May</td>
<td>Chinese-English Maths/Physics Journals</td>
<td>Interactive Direct Translation</td>
</tr>
<tr>
<td>Year</td>
<td>System</td>
<td>Description</td>
<td>Source(s)</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>1971</td>
<td>ALPS</td>
<td>Brigham Young University, Provo, UT (Corness, 14-15 November, 1986)</td>
<td>English-German French, Portuguese, Spanish</td>
</tr>
<tr>
<td>1977</td>
<td>MATEO</td>
<td>Developed by John Chandioux (Thouin, 5-6 November, 1981), (Chandioux, 8-9 March 1976)</td>
<td>English-French, Canadian Meteorological Center</td>
</tr>
<tr>
<td>1985</td>
<td>RUSLAN</td>
<td>Developed by Hajic J. at Charles University, Prague (J.Hajic, 1987)</td>
<td>Czech-Russian</td>
</tr>
<tr>
<td>1995</td>
<td>PONS</td>
<td>Developed by Helge Dyvik, Department of Linguistics and Phonetics, University of Bergen, Norway (Dyvik, 1995)</td>
<td>Norwegian-Swedish &amp; English-Norwegian Unrestricted domain</td>
</tr>
<tr>
<td>2000</td>
<td>English to Filipino MT system (Borra, 2000)</td>
<td>English-Filipino</td>
<td>Transfer approach using Lexically Functional Grammar (LFG)</td>
</tr>
<tr>
<td>2000</td>
<td>Turkish-English MT (Turhan &amp; Keyder, 1997)</td>
<td>Turkish-English</td>
<td>Hybrid Approach</td>
</tr>
<tr>
<td>2000</td>
<td>Bulgarian to Polish</td>
<td>Bulgarian-Polish</td>
<td>PONS approach</td>
</tr>
</tbody>
</table>
Indian Context

India is a culturally diversified multilingual country. Govt. of India has recognized 22 scheduled languages as official languages of Indian union. these 22 languages includes Assames, Bengali, Bodo, Dogri, Gujrati, Hindi, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Kannada, Kashmiri, Konkani, Maithili, Santhali, Sindhi, Tamil, Telugu and Urdu (TDIL).

The MT activity started in India in early 90’s, initial MT initiatives in India are attributed to IIIT, Kanpur, C-DAC, Pune, NCST(now C-DAC, Mumbai), Department
MT activities in India are broadly focused around following language pairs: English-Hindi, English-Indian Languages (IL), IL-IL, English-Kannada, Hindi-Kannada, English-Telugu, English-Bangla, Hindi-Punjabi, and Sanskrit-Hindi. Most of the MT projects are funded by Technical Development of Indian Languages (TDIL), Department of Information Technology (DIT), Ministry of Communication and Information Technology (MoCIT), Government of India, New Delhi (TDIL). In order to speed up the development task as well as encourage the collaboration amongst the various MT/NLP research groups, the DIT has initiated consortia approach for funding research projects. Under this approach consortia comprising of likeminded MT groups are formed with one institute as consortia leader and projects are sanctioned to consortia. Most of the MT projects are funded by Technical Development of Indian Languages (TDIL), Department of Information Technology (DIT), Ministry of Communication and Information Technology (MoCIT), Government of India, New Delhi (TDIL). DIT has setup four MT consortia E-ILMT (ANUVADKSH), E-ILMT (ANGLABHARATI approach) IL-IL MT (SAMPARK), and Sanskrit Machine Translation Consortia (TDIL). Besides government initiatives, some scholars are doing MT research out of their interest or funding from UGC, New Delhi, CIIL, and Mysore. Brief summary of the major Indian MT projects as compiled from (Naskar & Bandyopadhyay, 13-15 September 2005), (Badodekar, 2002), (Goyal, 2010), (Sanjay Kumar Dwivedi, 2010), (Sitender & Bawa, March 2012), and (Garje, 2013), is presented below. These systems have been classified as English to IL, IL-English and IL to IL MT systems.

A. English to Indian Languages (E-IL) MT Systems


Both systems have been specifically designed for translating English to Indian languages. Instead of designing separate translators for English to each Indian language, AnglaBharati uses a pseudo-interlingua approach in which English is parsed once and stored in an intermediate structure called Pseudo Lingua for Indian
Languages (PLIL). From PLIL, we need only generator specific to particular Indian Language to generate TL translation. Angla Bharati is rule based system, which used context free grammar like structures for English. These rules as learnt from corpus. Various versions based on AnglaBharati like Angla-Hindi, Angla- Marathi, Angla-Urdu, Angla-Kashmiri etc. were proposed and some have been developed and others are in development stage.

ANGLABHARTI-II overcomes shortcomings of earlier approach. It is based on EBMT approach; it uses Generalized Example-Base (GEB) for hybridization besides a Raw Example-Base (REB). Whenever addition of rulebase saturates i.e. it does not help in obtaining good quality translation, example base is augmented to enhance the performance of the system. The application of example-base and rulebase is applied in REB, GEB in sequential fashion till, system does not get success. This system also has provision for automatic pre-editing, paraphrasing, generalized & conditional multi-word expressions, NER etc. System is also equipped with module for error analysis and statistical language modelling for post editing purpose.

A.2 MANTRA (MACHINE assisted TRAnslating) by Applied AI group, C-DAC, Pune (AAI Group; C-DAC, Pune, 1997), (Group, mantra.aspx, 1999)

Initiated by Dr. Hemant Darbari, MANTRA is transfer based MT and uses TAG (Tree Adjoining Grammar) (Joshi, Levy, & Takahashi, February 1975), (Abeille, Schabes, & Joshi, 1990) to represent English and Hindi analysis. MANTRA translates English text into Hindi in a specified domain of personal administration specifically gazette notifications pertaining to government appointments, office orders, office memorandums and circulars. MANTRA is based on LTAG formalism. MANTRA system preserves the formatting of input word documents across the translation. The MANTRA approach is general, but the lexicon/grammar is limited to the language pair of the domain. Besides English-Hindi pair, MANTRA has been extended to Hindi-English and Hindi-Bengali language pairs. It is domain specific practically working MT system. It has flavors like MANTRA (Rajbhasha) for Department of Official Languages and MANTRA (Rajyasabha) for Rajya-Sabha Secretariat, the Upper House of Parliament of India. It translates the parliament proceedings such as Papers to be laid on the Table [PLOT], Bulletin Part-I, Bulletin Part-II, List of
Business [LOB] and Synopsis. Efforts are on to extend this system to other Indian Languages.

The MANTRA systems are operational and have been successfully implemented in Dept. of Official Languages and Rajya Sabha. The MANTRA has become part of “The 1999 Innovation Collection” on Information Technology at Smithsonian Institution’s National Museum of American History, Washington DC, USA. MANTRA case study can be found at web link: http://www.cwhonors.org/Search/his_4a_detail.asp?id=791.

A.3 MAT (Machine Assisted Translation (2002)) (Murthy, 2002)

This English-Kannada MT system was developed at Resource Centre for Indian Language Technology Solutions, University of Hyderabad. MAT is based on using the Universal Clause Structure Grammar (UCSG) formalism. MAT does the translation on clause level and constructs the target language (Kannada) sentence by reordering the TL dictionary mapped words. It operates in the domain of government circulars/notifications. Prof. Kavi Narayana Murthy played key role in the development of this system.


This English-Hindi transfer based MT system with special reference to weather narration domain has been designed and developed by Lata Gore. More details could not have been found on this system.


This Interlingua based MT project initiated at IIT Bombay in which UNL (Universal Networking Language) is used as representing the Interlingua. UNL is an International project of UNO aiming to develop UNL for major human languages of the world. This project has added new dimension to Indian MT activities and offers new avenues for MT research.

The system is reported to produce Hindi Output, in the domain of news, annual reports and technical phrases. MaTra has been developed by the Natural Language group of the Knowledge Based Computer Systems (KBCS) division at the National Centre for Software Technology (NCST), Mumbai (now renamed as C-DAC, Mumbai).

A.7 English-Telugu MT System

This system has been jointly developed at CALTS with IIIT, Hyderabad, Telugu University, Hyderabad and Osmania University, Hyderabad. The system uses Telugu-English lexicon of 42,000 words and has been reported to handle variety of complex sentences.

A.8 Shakti & Shiva (Naskar & Bandyopadhyay, 13-15 September 2005), (Bharati, et al., 2003)

Both are English-IL (Hindi, Marathi, Telugu) MT systems. These are jointly developed by LTRC, IIIT, Hyderabad, IISC, Bangalore and Carnegie Mellon University, USA. Shakti is based on RBMT, while Shiva is based on EBMT. Shakti uses Anal-Gen paradigm (analysis and generation) and is highly modular system. It has total 69 modules. 09 modules are responsible for English sentence analysis, while 24 are doing bilingual task such as TL root substitution, word reordering etc. Other remaining modules are used for TL generation. These systems use proprietary format SSF (Shakti Standard Format) to represent input sentence analysis.

A.9 ANUBAAD (Bandyopadhyay, 2004)

EBMT based MT for translating news headlines from English to Bengali. Prof. Sivaji Bandyopadhyay at Jadavpur University Kolkata has developed this system.


This is a TDIL, DIT, MoCIT\(^1\), Govt. of India, initiative. This EILMT project is offered by DIT to EILMT consortium of 13 institutes namely C-DAC, Pune, C-DAC, Mumbai, IIIT, Hyderabad, and I. I. Sc. Bangalore, IIT Mumbai, Jadavpur University

---

\(^1\) TDIL - Technical Development of Indian Languages, DIT - Dept. of Information Technology, MoCIT - Ministry of Communication and Information Technology.
Kolkata, Amrita University Coimbatore, IIIT Allahabad, Banasthali Vidyapeeth Banasthali and Utkal University Bhubaneswar, North Maharashtra University, Jalgaon, DDU, Nadiad(Gujrat), NEHU, Shilong (last three institutes were included in second phase in 2009) with C-DAC, Pune as Consortia Leader. The mandate is to develop Machine Translation from English to nine Indian languages in chosen language pairs namely English–Hindi, English–Bengali, English–Marathi, English–Oriya, English–Urdu, English–Tamil, English-Gujrati and English-Bodo in selected domain of tourism and health. The first phase of the project started in 2006 in which MT system for English–Hindi, English–Bengali, English–Marathi, English–Oriya, English–Urdu, English–Tamil, have been released. To further enhance the first phase MT system, the project has been continued in second phase which started in 2009. New language pairs English-Gujarati and English-Bodo have been introduced in second phase. The second phase system is named as ‘ANUVADKSH’. The present EILMT system is integrated with four Machine Translation (MT) engines namely, Anal-Gen Based MT (Parser and Generator), Example Based MT (EBMT), Statistical Machine Translation (SMT), TAG Based MT (Parser and Generator) which allow translating the documents from English to Hindi. TAG based MT engine also allows translating from English to Urdu, Oriya, Marathi, Bangla and Tamil. Second phase aims to enhance existing modules and tools of the MT system. Semantic LTAG Parser and Generator also have been proposed. Presently ANUVADKSH is scoped to the domains of Tourism, Health Care and Agriculture. The system has built in ranking system, which ranks the outputs of various MT engines and presents the best possible output to the end user. It can be operated online at the link [http://www.tdil_dc.in/components/com_mtsystem/CommonUI/homeMT.php](http://www.tdil_dc.in/components/com_mtsystem/CommonUI/homeMT.php).

A.11 *TRAIL IBM MT* (IBM)

Developed at IBM India Research Lab, New Delhi for English-Hindi Pair, initially it has been designed around EBMT paradigm, and later it has been shifted to SMT paradigm.
A.12 **ANUVAADAK** (MySmartSchool)

This is private commercial initiative by Delhi based firm Super Info Soft. Pvt. Ltd. general purpose English to Hindi MT tool it has built in dictionaries for various domains.

A.13 *Google Translate* (Och & Josef, June 2007)

This web based SMT based MT engine has been developed by Google Inc. It has 66 language pairs, which includes world’s most of the languages. English- Hindi and English-Urdu have been introduced since 2008. It also has other Indian Languages as well as Indian to Indian Language pair translation. Perhaps widely used, but quality in Indian context is only up to the level of basic message understanding. It is being refined every day. The system can also translate the websites online retaining the look and feel.

A.14 *Microsoft Bing Translator (2009)* (Microsoft, 2009)

The Microsoft’s Translation engine supports the English-Hindi pair. Details regarding the approach and architecture are not available. It is available online at the link [http://www.bing.com/translator/](http://www.bing.com/translator/)

B. Hindi to English


ANUBHARATI and its successor ANUBHARATI-II are used for Hindi to English as well as Hindi to other Indian Language pairs. ANUBHARATI MT platform has been proposed by Prof. R.M.K. Sinha for Hindi to English as well as other Indian Languages Translation based on EBMT approach. The core architecture has been revamped in second phase that uses Hierarchical Example base. Building hierarchical base for sister pair languages like Hind and other Indian Languages is lot easier than dissimilar language pair (Hindi and English, German etc.). The Hindi sentence is converted to `standardized form’ after shallow grammatical analysis this standardized form is matched with top level hierarchical example base, if exact match is not found input sentence segments(chunks) are matched in hierarchical example base, reordering of translated chunks is performed by matching sentence level example base. Manual
post editing facility is also made available. The system is reported to produce better output for sister language pairs.

B.2 *Hinglish* (Sinha & Thakur, 13-15 September, 2005)

This MT system does Hindi to English translation; it is extension of Anglabharati-II system in which additional layer has been incorporated to Anglabharati platform and works for pure (standard) Hindi. It is reported to have 90% satisfactory results.

C. Indian to Indian Languages (I-ILMT)

Most of the Indian initiatives are focused around Hindi, Punjabi, Bengali, south Indian languages (Telugu, Kannada, Tamil). For many systems Hindi is at either SL or TL side.

C.1 *ANUSAARAKA (1995)* (IIIT, Hyderabad)

Historically ANUSAARAKA is the first Indian MT initiative, it is a TDIL sponsored machine aided translation system, it started with IIT, Kanpur and later on shifted to CALTS, University of Hyderabad, and presently at LTRC, IIIT, Hyderabad. The prime focus of Anusaaraka is not mainly on Machine Translation; rather it is on providing language access between Indian Languages. It is based on the principles of Paninian Karaka theory and similarity of Indian languages, it essentially maps local word groups between the source and target languages. Input sentences are normalized in some cases to match with the NL analysers. Extra training is required understanding the output of Anusaaraka. Anusaaraka for different Indian languages (Punjabi, Bengali, Tamil, Telugu, Kannada and Marathi) to Hindi have been developed. However, we could not find details on Hindi-Marathi Anusaaraka. Anusaaraka is also available for English to Hindi pair and have reported to be doing well, these days.


This MT system is used for Bilingual Bengali-Assamese News Texts. It is based on Example-Based Machine Translation (EBMT) paradigm. It follows standard MT process flow, which requires pre-processing and post processing. Longer sentences are
fragmented into smaller parts at punctuation marks and backtracking is also used for further fragmentation of the sentence when exact match is not found.

C.3 Sanskrit-Hindi ANUSAARAKA (2007) (Bharti & Kulkarni, 2009)

Sanskrit-Hindi pair has been tried on ANUSAARAKA. This system is language accessor cum machine translation. This system has been developed by Prof. Amba Kulkarni, Dept. of Sanskrit Studies, University of Hyderabad.

C.4 Interlingua to Hindi generation (2007)

This project was initiated by Prof. Pushpak Bhattacharyya and Prof. O. P. Damani at IIT Bombay, Mumbai.


Developed at Punjabi University, this system is based on hybrid i.e. direct, rule based and statistical approach. The system has reported accuracy of 90%.

C.6 Sampark MT Engine (2009) (TDIL),

Sampark is I-ILMT consortia developed web based multiengine MT framework. It uses RBMT (Paninian Karaka theory) and SMT based MT engines. The member institutes include University of Hyderabad, Hyderabad, C-DAC Noida, Anna University, KBC Chennai, IIT Kharagpur, IISc Bangalore, IIIT Allahabad, Tamil University, IIT Bombay and JadHAVpur University, Kolkata. Presently Hindi to Punjabi bidirectional, Telugu to Tamil bidirectional, Tamil to Hindi, Telugu to Hindi, and Marathi to Hindi are available on the Sampark website (http://tdil-dc.in/sampark/web/index.php/content).


This multiengine hybrid (SMT and RMBT) Machine Translation system was developed at IIT Kharagpur by Chatterji S, Roy D, Sarkar S and Basu. Empirical results state that hybrid system yield better BLEU score (0.2275) as against individual experimental scores of 0.1745 and 0.424 for RBMT and SMT respectively.
This system has been developed under the guidance of Prof. G. S. Lehal, as part of the doctoral research. This system is based on direct approach. The system has reported accuracy of 95%.

2.4 Review of Grammar Formalisms (GF)

As proposed in the outline, HinMaT is based on Syntactic Transfer approach, hence parsing is inevitable and core part of MT life cycle. The parser heavily relies on the rule base, which is modelled using Grammar Formalism (GF). Grammar formalism is a mechanism used for formal, mathematical and computational modelling of natural languages. GF consists of set of permissible symbols (alphabet set), set of instructions and mechanism for manipulating symbols with the help of instructions. Instructions are the grammar rules and symbol manipulation mechanism is the derivation process for deriving a string belonging to that language. GF characterize set of infinite strings.

In order to choose suitable GF for our needs, we had to dig through history of GF. Based on the sentence analysis approach and representation style, there are two schools of thoughts in Computational Linguistics i.e. constituency and dependency.

Constituency school is inspired from ‘structuralism’ proposed by Ferdinand de Saussure (1857-1913). His preposition was that, elements of language are like pieces on a chess board and they only make sense, when you consider their role in relation to other pieces. Various grammar formalisms under this school are tightly tied with word positions of constituents which are mostly fixed hence these GFs are more suitable for fixed word order languages like English (Bharati, Chaitanya, & Sangal, 1995) and not suitable for representing free-word-order languages like Indian Languages, where constituents’ positions can be swapped without affecting the gross meaning.

This school treats a sentence as sequence of phrases, where a phrase can be recursively defined as composition of terminals (words in crude sense) and phrases. Phrases and their Intermediate level analysis are represented using special symbols called ‘Non-Terminals’ (NT). The sentence analysis drills through phrases to arrive at pure lexical level (all terminal symbols: Sentence surface) in recursive manner. A constituency analysis is represented using hierarchical tree structure called ‘derivation tree’ (pl. refer Figure 2.2 below).
Figure 2.2 Constituency Analysis: Derivation Tree for sentence “Stout boy kicked the ball”

The dependency school treats sentence as flat structure and assumes that words in a given sentence are bound to each other through binary asymmetric relationship called ‘dependency’. Further the dependencies can be layered into different levels like morphological, syntactic and semantic. Following Figure 2.3 shows dependency analysis of Hindi sentence, “राम ने मीठा आम खाया।”.

Figure 2.3 Dependency Analysis: Hindi sentence “राम ने मीठा आम खाया।”

Due to its flat treatment to sentences, Grammar formalisms under dependency school are not coherent and we can find different kinds of dependency notations and DG flavors. They operate at the surface i.e. lexical level due to which they are most suitable for free-word-order languages. E.g. Consider Hindi sentence राम ने आम खाया[Ram ate mango], the constituents can be swapped in any order like आम राम ने खाया, खाया आम राम ने, without affecting or changing the meaning this is not possible in English. However this school imposes many constraints on its’ grammar formalisms, which include Single Headedness (uniqueness) i.e. a dependent can have
only one Head (some DGs allow multiple headedness: e.g. राम ने खाना खाकर पानी पिया [Ram drank water after eating food]), Projectivity (Adjacency) i.e. dependents and their head should be adjacent i.e. no crossing edges in dependency analysis representation. The last constraint imposes Linear Ordering of words in the sentence. Parsing procedures are available even for parsing non-projective constructions (R. McDonald, 2005), (Carbonell, 1987), (Tapanainen & Järvinen, 1997).

Different grammar formalisms have been proposed under each school. The representative formalisms under each school are described in brief in the following section.

2.4.1 Constituency School grammar formalisms

The primary research in this school dates back to 19th century. Many scholars have contributed to evolution of new grammar formalisms to this family including Context Free Grammar(CFG) (Chomsky N., 1986) (Chomsky N., Feb 1957), Transformation grammar, Montague Grammar (Barbara, 1976), Functional Unification grammar (Kay, 1984), Lexical Functional Grammar(LFG) (Kaplan & Bresnan, 1982), Generalized Phrase Structure Grammar(GPSG) (Gazdar, Klein, Pullum, & Sag, 1985), Head Driven PSG(HDPSG) (Pollard & Sag, 1994), Tree Adjoining Grammar (Joshi, Levy, & Takahashi, February 1975), Left Associative Grammar(LA) (Hausser, 1988), (Pawar B. V., 2001), etc. Detail discussion on each of the formalisms is out of scope of this thesis, but still we have briefed few popular formalisms under this school from various sources.

2.4.1.1 Context Free Grammar (CFG)

The CFG by Noam Chomsky (Chomsky N., Feb 1957) is most popular and much discussed grammar formalism till date. CFG is the base formalism for most generative grammars. It is the first manifestation of structuralism in linguistics. CFG has been computationally very well tested and understood GF. Good amount of parsing algorithms like LR(k) (Knuth, November 14, 1997), Earley style (Earley, 1970), CYK (Hopcroft, Motwani, & Ullman, 2001) (Cocke, John, Schwartz, & T, 1970), (Younger, February 1967), (Kasami, 1965), Tomita (Tomita, July, 1984) are available for research community. CFG is defined as,
CFG is defined as 4-tuple (\( \Sigma, NT, T, P, S \)),

Where,

\( \Sigma \) is called alphabet set (Lexicon) containing all possible symbols in the grammar.

\( NT \): is set of non-terminal which is special category (immediate constituent structure).

\( T \): is set of all terminal (surface) symbols.

\( P \): is set of rules (productions) and

\( S \): is specially designated non terminal ‘start symbol’ from which every sentence analysis starts.

Formal methods for deriving particular surface structures are known as ‘derivation’ in CFG. The main claims of CFG as described in (Ivan A. Sag, 2006) states that:

1) Parts of sentences (larger than single words) i.e. phrases are linguistically significant units, which play a role in determining meaning, pronunciation, and/or the acceptability of sentences.

2) Phrases are contiguous portions of a sentence (no discontinuous constituents).

3) Two phrases are either disjoint or one fully contains the other (no partially overlapping constituents).

4) Internal structure (contents) of phrase depends only on what kind of phrase it is (that is, the label on its top node), not on what appears around (position of phrase) it, so to say they are ‘context free’.

But it is very common to witness counter examples in most Natural Languages, against these claims, hence CFG’s are not fully adequate for modelling syntax of natural languages. But one can’t deny the fact that they gave birth to phrase structure grammars (PSG). Original CFG formalism does not talk about applying constraints (agreement) on constituents nor of defining features (syntactic or semantic) on phrases. This fact give lead to other scholars who worked on these aspects and came up with variety of new grammar formalisms. A typical CFG fragment (which is also called universal) is portrayed below:

\[ \Sigma = \{NP, VP, det, N, ADJ, v, Aux, Adverb\} \]

\[ NT = \{NP, VP, N, ADJ\}, T = \{det, v, Aux\} \] and set of production \( P \) as;
2.4.1.2 Lexical Functional Grammar (LFG)

Kaplan and Bresnan (1982) proposed the LFG formalism to PSG family. LFG postulates three distinct levels\(^2\) of representations which are interrelated, these levels are Lexical structure, Constituency structure (c-structure) and Functional structure (F-structure). The c-structure is specified by using CFG rules while f-structure is associated with every symbol of CFG production. Following sections explains various levels of LFG representation with suitable example.

**Lexical structure:** Lexical entries specify meaning of lexical item, its argument structure and grammatical function. An entry for verb ‘beat’ may look like:

beat: verb

\[(\uparrow \text{PRED}) = '{\text{meaning of beat}} <\text{SUB}, \text{OBJ}>'\]

Boy noun

\[(\uparrow \text{PRED}) = '\text{boy}'\]

\[(\uparrow \text{Num}) = \text{Singular}\]

‘\(\uparrow\)’ is a meta variable under which this entry is found.

**C-structure:** Constituent structure encodes linear order, hierarchical grouping and syntactic categories of constituents (Neidle, 12-13 Augst, 1996). C-structure is represented using CFG rules.

**F-structure:** Feature structures are encoded using AVM (Attribute Value Matrix) notation. They represent attribute-value pairs, where value of attribute can be another feature structure or symbols. In the former case it is called as ‘grammatical function’

---

\(^2\) Most of the LFG literature does not consider lexical structure as level of representation rather they advocate that the lexical information encoded in this level, is included in the lexicon.
and later is simply called as ‘feature’ (Laura Kallmeyer, 2007). Following Figure 2.4 shows C-structure and F-structure.

Our main objective is to link the C-structure and F-structure, which is achieved by using special notation as shown below.

\[
\begin{align*}
S & \rightarrow \text{NP} \quad \text{VP} \\
& \quad (\uparrow \text{sub}=\downarrow) \quad (\uparrow=\downarrow) \\
\text{NP} & \rightarrow \text{det} \quad \text{N} \\
& \quad (\uparrow=\downarrow) \quad (\uparrow=\downarrow) \\
\text{VP} & \rightarrow \text{V} \quad \text{NP} \\
& \quad (\uparrow=\downarrow) \quad (\uparrow \text{obj}=\downarrow)
\end{align*}
\]

‘\(^\uparrow\)’ refers to meta variable associated with f-structure of non-terminal symbol (mother) on the left side (LHS) of CFG rule, while ‘\(^\downarrow\)’ refers to right side (RHS) symbol (daughter) of CFG rule.

![Figure 2.4 C-structure and F-structure representation](image)

The equation \(\uparrow \text{sub}=\downarrow\) states that there is a feature named ‘sub’ in the f-structure of mother whose value is same as that of daughter in right hand side, while \(\uparrow=\downarrow\) states that value of f-structure in mother is same as that of daughter. LFG is capable of modelling long distance dependencies. For Well-formedness and correctness of F-structure representation, coherence and completeness conditions are enforced on LFG.

2.4.1.3 Functional Unification Grammar (FUG)

Functional Grammar is important theory about organization of Language in the history of linguistics in general and specifically in computational linguistics. Early conceptualization of functional grammar was due to Dik Simon (Simon, 1997). Martin Kay (Kay, 1979), (Kay, 1984) further elaborated this theory and added new
dimensions to this theory. FUG serves as underlying formalism for other GFs like LFG, HDPSG, TAG and CC grammar etc. are based on FUG. According to Kay (Kay, 1979), functional grammars is different from conventional linguistics theories in three ways. First, it presents the functional view of grammar so that it can function as part of model of language production and comprehension i.e. it can be interpreted and analysed by abstract machines whose job is to model syntactic processing. Secondly, formalism ascribes a functional description to every word/phrase/sentence. This description is different from structural description. It stresses more on function that a part plays in a whole. Role terms like subject, object and modifier are primary to grammatical category labels like S, NP, and VP etc. Thirdly, properties that distinguish among logically equivalent sentences will have equal importance with properties that they share. In this view functionalist terms like topic, focus, given etc. will be at par with subject, predicate, positive, negative etc.

In FUG, functional notions play essential and fundamental roles at different levels of grammatical organization. Many of the rules and principles of FUG are formulated in terms of following functional notions. Three types or levels of functions are distinguished, In Kay’s words they are:

**SEMANTIC FUNCTIONS** defines the roles such as Agent, Patient, Recipient, etc. that participants play in states of affairs, as designated by predications.

**SYNTACTIC FUNCTIONS** (Subject and Object) which define different perspectives through which states of affairs are presented in linguistic expressions.

**PRAGMATIC FUNCTIONS** (Theme and Tail, Topic and Focus) hold the informational status of constituents of linguistic expressions across the sentences. They relate the embeddings of the expression in the current discourse as determined by the status of the pragmatic information of verbal communication between Speaker and Addressee.

Functional description describes the properties of words (objects) which are otherwise not given more importance in CFG. These are modelled with the help of feature structures, which are represented using attribute-value matrix notation. Property (attribute) of object (word) is called as feature, values of feature may be constrained by predefined atomic value set or may be unconstrained (unbounded: open set). A
feature whose value is atomic is called ‘atomic feature’. Theory of features allow values of feature to be defined as another feature or feature structure in recursive manner such features are called as ‘complex features’. Due to functional view of the grammar any number of functional categories can be defined and associated with objects (words). All this information is stored in the lexicon so the syntactic aspects are captured and stored in the lexicon thus simplifying syntactic processing components and hence reducing the computational complexity. Set of features pertaining to an object define the feature structure for that object. FUG also prescribes the theory of unification of feature structures, to enforce agreement, control and syntactic/semantic compatibility between constituents at various levels ranging from word level to phrase/clausal levels. Details of unification process are discussed in (Wintner, 2008) so we can say that FUG extend CFG and is an important milestone in the journey of GFs.

2.4.1.4 Generalized Phrase Structure Grammar (GPSG)

GPSG was proposed by G. Gazdar, G. K. Pullum et al. (G.Gazdar, Klein, Pullum, & Sag, 1985). GPSG was introduced as alternative to Transformation Grammar (TG), which had dominated the linguistics and research field. GPSG belong to family of unification based grammars. GPSG is monostatal formalism which uses CFG as base syntax.

2.4.1.5 Head Driven Phrase structure Grammar (HDPSG)

HDPSG is a unification based theory of grammar in which structures satisfy constraints for all forms, all at once. Originally HDPSG was proposed by Pollard & Sag (Pollard & Sag, 1994), (Ivan A. Sag, 2006). HDPSG relies on two components:

i) Structured representation encoded using feature structure representation of grammatical categories

ii) Constraints specification through feature structure. Basic unit of linguistic data representation in HDPSG is ‘sign’, which is feature structure containing phonological, morphological, syntactic and semantic information about the lexical unit. A feature structure (f-structure) is collection of attribute-value pairs, where each attribute value may be another feature structure (recursive). Feature structures may be shared among
various categories. Sharing of f-structure implies constraints (Meurers, Levine, & Detmar, 2006).

Following Figure 2.5 shows a sample phrasal construction from (Meurers, Levine, & Detmar, 2006) represented using Attribute Value Matrix (AVM) notation using tree. HDPSG permits refinement of grammatical categories classification. It also incorporates ideas from GPSG. Other central ideas pertaining to HDPSG include, concrete surface oriented structures, geometric representation, locality, ID/LP principles, Lexical and Hierarchical cross classification of grammar constructions.

![HDPSG Phrasal construction representation](image)

**Figure 2.5 HDPSG Phrasal construction representation**

2.4.1.6 Categorial Grammar (Steedman, 1987), (Morril, 1995), (Curry, Feys, & Craig, 1958)

The basic ideas of Categorial grammar date back to work by Kazimierz Ajdukiewicz (in 1935) and Yehoshua Bar-Hillel (in 1953), this was further taken up by Joachim Lambek in 1958, when he introduced a syntactic calculus that formalized the function type constructors along with various rules for the combination of functions (Categorial_grammar).

Categorial grammar is based on the principle of compositionality, as such it does not have standard notation. It has a lexicon and set of types (also called categories) which
are assigned to basic symbols (lexical entries) and set of inference rules. Rules can be applied in two fashions i.e. ‘forward application’ or ‘backward application’. This formalism is more simplified as it moves the complexity from rules to lexical entries. Categorical Grammars are more tightly coupled with semantics particularly lambda calculus and there is one to one relationship from syntactic and semantic constituents. Categorical grammar defines 5 core rules for various composition operations, which are:

i) Rule Application:

   Forward Application: A/B + B = A

   Backward Application: B + A\B = A

ii) Composition: A/B + B/C = A/C

iii) Coordination: A CONJ A’ = A”

iv) Type raising: A = X/(X\A)

A, B, A/B in above rules are types (categories).

Consider derivation of English sentence “Peter loves Jenny” is given below:

Lexicon: {Peter = np, Jenny = np, loves = (s\np)/np}

Derivation Steps: np (Peter) (s\np)/np (loves) …… no rule can be applied

np (Peter) (s\np)/np (loves) np(Jenny) ……. Now apply forward rule

   np (Peter) s\np (loves Jenny) …………….. Now apply backward rule

   s (Peter loves Jenny) ……………….. Derivation complete

Categorical grammar can handle the both constituent conjunctions (NP and NP, VP, NP, VP and VP) as well as non-constituent conjunction cases. Consider the NP, VP and VP conjunction case English example (John likes Mary and dislikes Bob) below:

Constituent Conjunction Case
Handling of non-constituent conjunction case is shown below; it is handled with the help of Type Rising operation.

Non- Constituent Conjunction Case

2.4.1.7 Tree Adjoining Grammar (TAG)

Tree Adjoining Grammar (TAG) proposed by Prof. Arvind Joshi (Joshi & Schabes, 1997), (XTAG Research Group, 2002) is considered to be leading grammar formalism in Generative Enterprise (Bharati, Chaitanya, & Sangal, 1995). TAG models lexical categories using hierarchical tree structures. Trees can be combined using two operations viz. Adjunction and Substitution. In simple words essential parts (arguments/valency) of sentence are added to sentential tree using substitution, while optional parts (modifiers) are adjuncted on modified item’s tree. Every tree contains a specially designated node called ‘anchor’. Tree is lexicalized at anchor (marked with diamond symbol) using lexical item; such a tree is called as Lexicalized TAG (LTAG) tree. Constraints can be defined on LTAG tree nodes for controlling adjunctions (NA,
The elementary LTAG trees are divided into two classes viz. Initial (alpha-α) tree & adjunction (Beta-β). Leaf nodes in elementary tree is either marked with anchor (Φ) or substitution constraint (↑↓ symbol) or star (* symbol) i.e. foot node. Substitution operation can be performed using alpha tree only, while adjunction can be performed with the help of beta tree only. Beta tree gets adjuncted on elementary trees (alpha or beta). Sentence analysis in LTAG involves combining the tree structures associated with lexical items (words) either by Adjunction or Substitution operations, in a special tree called sentential tree (generally tree anchored at verb). These operations are performed on Nodes of elementary trees which are marked with special constraints for adjunction and substitution. After combining all lexical items the resulting hierarchical structure results into tree data structure for sentence, which is also called as ‘parse tree’ or ‘derived tree’ for the sentence. The yield of this tree i.e. leaf nodes (words) in left to right order represent the analyzed sentence. Another plus point with TAG is that it can recognize mildly sensitive grammars. Feature structure can be defined on the nodes of LTAG tree, such LTAG is called as F-LTAG. The following diagram Figure 2.6 shows the various trees in LTAG along with derived tree.

![Figure 2.6 LTAG Trees & Derivation](image)

2.4.1.8 Left Associative Grammar (LAG)

This grammar formalism was proposed by Ronald Hausser (Hausser, 1988). In his own words, “a grammar that uses derivation order of left associative combinations is called the Left Associative Grammar (LAG)”.

---

3 NA- Null Adjunctioon, OA-obligatory Adjunction, SA- Selective Adjunction
4 Foot node is marked with same label as that of the root node of beta tree
Formally, it is defined as 7 tuple \(<W, C, LX, CO, RP, ST_s, ST_f>\), where,

\(W\) is a finite set of word surfaces.

\(C\) is finite set of category segments. Category is a sequence of non-empty Category Segments of \(C^+\). E.g. N, D, A, V are category segments of C then \((N \ D \ A \ V)\) is category of \(C^+\).

\(LX\) is Lexicon comprising finite subset of \((W \times C^+)\). A Categorized word is an element of \((W \times C^+)\). E.g. If gave belongs to \(W\) is a word and \((N \ D \ A \ V)\) is category of \(C^+\) then, \(+ [\text{gave} \ (N \ D \ A \ V)]\) is a categorized word of \((W \times C^+)\).

\(CO = (c_{0}, \ldots c_{n-1})\) is a finite sequence of total recursive functions from \((C^* \times C^*)\) in \(C^* \cup \{\#\}\), called Categorical Operation.

\(RP = (r_{p0}, \ldots r_{pn-1})\) is an equally long sequence of subsets of \(n\) called Rule Packages. Left Associative rule \(R_i\) is defined as \((C_{o_i}, R_{p_i})\).

\(R_i: [\text{CAT-1}, \text{CAT-2}] \Rightarrow [r_{p_i}, \text{CAT-3}]\)

\(\text{CAT-1}\) Category of current Sentence Start, \(\text{CAT-2}\) Category of Next word, \(\text{CAT-3}\) Category of new Sentence Start, \(r_{p_i}\) is Rule Package of rule \(i\) (Rule package is set of Rule names)

\(ST_s= \{(r_{p_s}, c_{ats})\ldots\}\) is a set of initial states \(r_{p_s}\) is called a start rule package, \(c_{ats}\) belongs to \(C^+\).

\(ST_f= \{(r_{p_f}, c_{atf})\ldots\}\) is a set of final states \(r_{p_f}\) belongs to \(RP\) package, \(c_{atf}\) belongs to \(C^*\).

Strength of LAG lays in the fact that it can be parsed in linear time order i.e. \(O(n)\) and \(O(n^2)\) in case of ambiguity. Use of Features is also possible. The natural Left to right order for processing, makes it lucrative for speech processing. Sample derivation tree in LAG is shown in following Figure 2.7
2.4.2 Dependency School grammar formalisms

In most of the literature that we came across, we found that this school of thought has oldest history which dates back to 500 B.C. (Panini), references of this tradition of sentence analysis could also be found in Greek and Roman Linguistic Tradition (100 B.C. – V c. A.D.), Arabic Linguistic Tradition – VIII c (Sibawaihi, d. ca. 793) and European Linguistics (Modistae, ca. 1260–1310). Surprisingly there is no mention of DG like framework from mid of 14th century till 19th century though the notions of modifier-modified was coined by French grammarian Claude Buffier (early 18th c.). In 1959, there is a mention of DG work by Lucien Tesnière ‘, whose book ‘El´ements de syntaxe Structurale’ (Syntax Elements) was published posthumously. His approach to syntax can be described with the help of notions like connexion, autonomous syntax, verb centrality, stemmas, centripetal (head-initial) and centrifugal (head-final) languages, valency, actants and circonstants, and transfer. Notion of stemma is central for his representation. Stemma is tree like structures. Connexion (connection) is binary dependency relation, while Junction (jonction) is relationship between dependents of same head or heads of same dependent (coordination like And, Or etc.). Transfer (translation) construct denotes change in POS category of a word at run time due to context. In his work Tesnière has listed examples from 60 languages. His work is amongst the first attempt to model syntactic description, importantly he introduced the notion of nucleus. According to him, a nucleus may consist of a single word or multiple words, which may be discontinuous. Nucleus is used as a descriptive primitive instead of the word, i.e. a nucleus is an element that can appear as a node in
the functional description. In his formulation, Tesni’ere ignored the word order. This is considered as the only limitation of this theory. Important aspects of Dependency Grammar (DG) are explained in the following section.

2.4.2.1 Dependency Relations & Constraints

Basic notion of dependency relation is based on relationship between words of a given sentence. A → B means word A governs word B or word B depends on word A. Other terms like ‘governor’, ‘modified’ and ‘regent’ are also used synonymously for marking ‘head’ of dependency relation, while ‘modifier’, ‘dependent’ & ‘subordinate’ are used to mark ‘dependent’. Basis for defining ‘dependency’ may be based on syntactic or semantic properties, or both. Based on the levels of abstraction DGs may be classified as mono-stratal and multi-stratal. Initial attempts to study mathematical properties of DG were made by Hays (Hays, 1964) and Gaifman (Gaifman, 1965). After that Robinson (Robinson, 1970) studied the mathematical properties of DG and formulated following four axioms:

i) One and only one element is independent (generally the root of sentence).

ii) All others depend directly on some element.

iii) No element depends directly on more than one (other) element.

iv) If A depends directly on B and some element C intervenes between them (in the linear order of the string), then C depends directly on A or B or some other intervening element.

The first three axioms are sufficient to define the well-formedness of DG and states that DG is a tree structure. Axiom iii) talks about ‘single headedness’ while axiom iv) talks of ‘projectivity’ i.e. disallows crossing edges in the dependency tree.

2.4.2.2 Representation Variants

Sentence analysis in DG framework is represented using dependency graph or also referred as dependency tree. Different variants of this representation are observed in the DG literature. In Tesni’ere’s terminology, dependency graphs (or dependency trees) can be represented using two kinds of stemmas viz. real stemma and virtual stemma. But currently the trend is to combine these two into single representation (pl. see Figure 2.8)
In other representation, the arcs are drawn just above the words of the sentence (pl. refer Figure 2.9 below). As such there is no clear cut convention regarding drawing the arc directions (arrow) which may be drawn from ‘Head’ to ‘dependent’ or ‘dependent’ to ‘Head’. But widely accepted convention is to draw arrow of arc towards dependent side.

In some variants, arcs are labelled, with dependency relation mnemonic to denote dependency function that defines the functional role.

Dependency structure can be modelled using directed graph G, where V is set of nodes and E is set of edges connecting two nodes. Nodes are also labelled with word forms (or POS categories). The directed edges of G may also be labelled with respective dependency relations. Formal properties of Dependency graph are characterized by following principles:

i) Antisymmetric
If A → B i.e. A governs B, then B → A is not true (or possible). E.g. horse → white then white → horse is not possible simultaneously.

ii) Antireflexive

If A → B then A ≠ B i.e. A can’t govern itself. In other words graph can’t have self loops.

iii) Antitransitive

If A → B and B → C then A → C is not true. E.g. Consider the noun phrase very beautiful girl, here beautiful → very and girl→ beautiful hold but girl→ very is not possible and also it is grammatically incorrect.

iv) Labelled arcs

For all ‘→’ (arcs), every ‘→’ must have a label ‘r’.

2.4.2.3 Comparison with PSG School

Hays (Hays, 1964) and Gaifman (Gaifman, 1965) tried to study the equivalence relationship between CFG and DG and concluded that CFG and DG are not strongly equivalent, it is not possible to map any CFG to corresponding DG however the reverse is true. The detail proof can be found in (Debusman, 2000). DG has several advantages (Rambow & Joshi, 1997) over PSG, like DG representation has close connection to semantic representation, some typological regularity can also be studied easily however lack of constituent makes analysing coordination’s difficult. Also, DG treats modification of constituent structure (c-structure) and individual words i.e. clausal modification versus individual word modifications in same fashion, which may impose constraints on exact analysis.

Various GF flavours of this school are described in brief in the following section.

2.4.2.4 Meaning Text Theory (MTT)

MTT (Mel'cuk, 1987) is based on the idea that Language is a means of representing meaning with the help of text. MTT is characterized by modelling language as a correspondence between meaning and texts/sounds, the central position of the lexicon, the privileged position of semantics over syntax, and the use of dependency representation in syntax (WikiAuthor:http://meaningtext.net/wiki). MTT represents Language as 7 strata of representation, and uses rules for mapping unordered dependency trees of surface-syntactic representations onto the annotated lexeme
sequences of deep-morphological representations. Each representation layer ensures correspondence between two adjacent levels. Discontinuities are accounted for by global ordering rules. The layers include:

1) Semantic representation
2) Deep syntactic representation
3) Surface syntactic representation
4) Deep morphological representation
5) Surface morphological representation
6) Deep phonological representation
7) Surface phonological representation

2.4.2.5 Word Grammar (Hudson 1990)

Word Grammar (WG) (Hudson R. A., 1990), (Hudson R. A., 2000) is broader framework covering morphology, syntax and semantics. The main ideas of WG can be found in most of the other theories in linguistics like Auto lexical Syntax, Categorical Grammar, Cognitive Grammar, Construction Grammar, Functional Grammar, Head-driven Phrase-Structure Grammar, Lexicase, Lexical-Functional Grammar, Link Grammar, Systemic Functional Grammar and Tree-Adjoining Grammar. The Syntactic structure of sentence consists of network of dependencies (multiple headed dependencies) and structure of grammar is a network of word sized concepts (nodes), the nodes are connected to each other through dependency relations. WG is unique in the sense that it is the only theory which uses bundle of features from other theories. In general WG is based on graphs instead of trees. Words in the sentence are linked using dependency relations, in which multiple heads are allowed Main ideas of WG as excerpted from (Hudson R.), are presented below:

1. It is based on network of knowledge that links concepts associated with word to its morphological, syntax and meaning.
2. WG uses mono-stratal (single layer) representation.
3. It uses word-word dependencies and no phrases.
4. It specifies the grammar functions using arc labels.
5. It makes use of features to handle inflectional contrasts as encoded in agreement rules.
6. It uses ‘default inheritance’ to handle contrasts in various patterns (basic, underlying as well as exceptions and transformations).

A typical WG sentence analysis representation using WG graph is shown in following Figure 2.10.

![Figure 2.10 WG Graph](image)

2.4.2.6 Link Grammar

This GF was formulated by Daniel D. K. Sleator and Davy Temperley (Sleator & Temperley, 1991), (Sleator & Temperley, 1995). It is treated as derivative of DG. It has additional two parameters over DG i.e. directionality and distance. Directionality refers to ‘left’ or ‘right’ direction of the current word under consideration, while distance distinguishes specific link requirement on either side. Dependency between two words is specified using ‘link’ relation. Every word (terminal symbol of grammar) in the grammar specifies its link requirements. Sequence of words is defined as a valid sentence in the grammar, if and only if, we can draw links between these words such that links satisfy following conditions:

- **Planarity**: This condition states that no two links cross each other.
- **Connectivity**: All words are linked with ‘link’ (no isolated words).
- **Satisfaction**: Linking requirement of each word in the sentence is satisfied.

Set of links that prove that sentence is valid sentence is called as ‘linkage’. Complete Parsing framework is available for Link Grammar at URL
Figure 2.11 below shows correctly parsed sentence.

![Figure 2.11 Link Grammar Parse example](http://www.cs.cmu.edu/~sleator/paper/node1.html)

Wrong sentence generates incomplete parse as demonstrated in Figure 2.12 below. The link requirement of word ‘Mary’ is not satisfied so does the requirement of word ‘cat’.

![Figure 2.12 Incomplete Parse (Mary*)](http://www.cs.cmu.edu/~sleator/paper/node1.html)

2.4.2.7 Functional Dependency Grammar (FDG)

The Functional Dependency Grammar (FDG) framework was proposed by Tapanainen and Järvinen (Tapanainen & Järvinen, 1997). Supposedly it is the first implementation of Tesnière’s DG. It consists of conceptually two different components i.e. regular language based analysis and structural dependency graph. It uses following assumptions:

1. Basic sentence structure of analysis is not a word but nucleus.
2. Nuclei of the sentence are connected with directed arcs (dependencies) called connexions.
3. Every nucleus has one and only one syntactic head thus resultant dependency graph is a tree
4. Variation of word order does not affect the structural analysis as long as syntactic function of word does not change. Thus it supports non-contiguous nucleus, which means FDG is non-projective in nature
5. Syntactic structure is motivated by semantic interpretation than word order constraints thus there is parallelism in syntax and semantics.

Tapanainen and J’arvinen (Tapanainen & J’arvinen, 1997) have also proposed a parser for parsing non-projective constructs.

2.4.2.8 Functional Generative Description (FGD)

Machine Translation has been prime motivation of this theory FGD was given by Petr Sgall, Eva Hajičová, Jarmila Panevová (Sgall, Hajičová, & Panevová, 1986). This theory uses ‘stratification’ approach with dependency framework. It uses layers of meaning. It also has generative and translation components. System of layers, valency theory, topic focus articulation and anaphora/co-reference are fundamental components of this theory. It uses more detailed layers than other multi-stratal approaches. It uses following layers:

- **t-layer** (tecto grammatical layer): captures deep syntax comprising language meaning in a form of a dependency tree. Valency, dependency, topic-focus articulation are the core concepts of this layer.
- **a-layer** (analytical layer): Used to capture surface syntax in the form of a dependency tree.
- **m-layer** (morphological layer): captures morphology triples: word form, lemma, tag.
- **w-layer** (word layer): captures individual words and punctuation marks in the form of a simple string.

There is one-to-one correspondence between individual symbols of the w- and m-layer.

The biggest contribution FGD is development of Prague Tree Bank (PDT), which is very important tree bank for dependency school. The details on PDT can be found (PDT guide) at [http://ufal.mff.cuni.cz/pdt2.0/](http://ufal.mff.cuni.cz/pdt2.0/)
2.4.2.9 Constraint Dependency Grammar (CDG)

Maruyama (Maruyama, 1990), (Maruyama, 1990) introduced CDG to DG family. The basic idea is to impose certain constraints on dependency relations so as to minimize the unwanted dependency relations in dependency parses. In CDG, every word specifies two primary roles, it’s ‘Governor’ (Head word, who governs current word) and it’s ‘Need’ (dependents of a head word in classical DG sense)\(^5\), the details are described in following section. Theoretically CDG is defined as 5 tuple (\(\Sigma, R, L, C, T\)), where \(\Sigma\) is finite set of lexical categories e.g. determiner, adjective etc., \(R\) is uniquely named roles (e.g. Governor, Need), \(L\) is set of labels (e.g. subject, object), \(C\) is constraint formula to verify grammatical correctness of input & \(T\) is table specifying which roles are supported by a lexical category, permissible set of labels for each role & lexical category. The number of roles generally denoted by \(p\) (per word) is called as the degree of CDG.

The roles values for a role ‘\(r\)’ are specified as \(l-m\), where \(l\) is a label specifying syntactic function, what a word will take, while in role ‘\(r\)’ and \(m\) is position that current word \(w\) is modifying when word \(w\) appears in the sentence with syntactic function ‘\(l\)’ for role ‘\(r\)’. Let \(L(G)\) denote language generated by CDG \(G\). A sentence ‘\(S\)’ belongs to \(L(G)\), if for every role of each word \((w_i)\) has one assigned role value such that constraint \(C\) is satisfied. A Sample CDG for sentence, ‘The boy eats pizza’ is shown in following Figure 2.13

| 1 | The Determiner | 2 | boy Noun | 3 | eats Verb | 4 | pizza Noun |
|---|---|---|---|---|---|---|
| G=det-2 | G=det-2 | G=root-3 | G=det-2 |
| N1=blank-nil | N1=blank-nil | N1=s1-2 | N1=blank-nil |
| N2=blank-nil | N2=blank-nil | N2=s2-4 | N2=blank-nil |
| N3=blank-nil | N3=detprt-1 | N3=blank-nil | N3=blank-nil |

**Figure 2.13 Sample CDG Fragment**

In this example, the degree of grammar is 4 because we have considered roles \(G\), \(N_1\), \(N_2\), and \(N_3\) (one governor role and three need roles). Role value of \(G\) in Dog is sub-3, which means sub is label (syntactic function) it takes in this role and 3 is the position

\(^5\) \(G\)-denotes governor role, \(N_i\)- denote the \(i^{th}\) need of current word
of its’ modifiee (head) ‘eats’. CDG parsing problem can be viewed as constraint satisfaction problem, where constraints are rules and solutions are parses. CDG allows crossing of links (non-projectivity), various parsing algorithms are available for parsing CDG (Maruyama, 1990).

2.4.2.10 Extensible Dependency Grammar (XDG)

XDG was proposed by Ralph Debusmann (Debusmann, Duchier, & Kruijff, 2004) (Debusmann, 2007) and it is a descendent of Topological Dependency Grammar (TDG) (Debusmann & Duchier, 2001). XDG finds its position between mono-stratal (Pollard & Sag, 1994) and multi-stratal (Mel'cuk, 1987), (Kaplan & Bresnan, 1982) approaches. It attempts to generalize two aspects of TDG i.e. number of dimensions and set of principles, which are fixed in TDG. TDG has two dimensions i.e. Immediate Dominance [ID] and Linear Precedence [LP]. XDG allows extending the number of dimensions and set of principles. XDG takes advantage of flexibility in DG, declarativity in Model theory and achieves modularity from parallel architecture. XDG is represented using multigraph model where models of XDG are tuples of dependency graphs. Each component dependency graph is called dimension. XDG allows an arbitrary number of dimensions. All dependency graphs (roughly one per dimension) share common set of nodes. Each dependency graph has following specific properties as excerpted from (Debusmann, Duchier, & Kruijff, 2004) (Debusmann, 2007):

1. Every node in the graph is drawn with circle and is associated with a index (1, 2, 3 etc.) to mark word positions. All nodes are connected by labelled edges (dependency relations) and each node is associated with a word in the sentences. Nodes and words are connected with words in the sentence by dotted vertical lines called ‘projection lines’. The actual written words are down below indices.

2. Each node is associated with attributes arranged in attribute-value matrices drawn below words. Attributes may specify various syntactic/semantic specifications including in-attributes and out-attributes referring to incoming edges and out going edges. They can also be used to indciate valency(argument structure) requirements of a node. Attribute value specification can make use of regular expression notations like ‘?’ (at the most
one), ‘!’ (exactly one), ‘*’ (zero or more). Following Figure 2.14 shows a sample dependency graph for sentence, ‘Mary wants to eat spaghetti today.’ as described in (37) for syntactic analysis dimension.

Figure 2.14 Sample XDG dependency graph

XDG can have such different DGs for each dimension (syntactic, semantic and syntactico-semantic). Additionally for defining XDG, we require principles and a lexicon. For a given XDG, we need to define various well-formedness conditions called principles like Tree principle (to state the DG is a tree), valence principle (all requirements of a word for the dimension under study are met), projectivity principle (tree does not have any crossing links) etc. As such there is no limit on number of principles. A lexicon in XDG stores set of records called ‘lexical entries’. Each entry is indexed with lexical anchor (actual word), attribute specification for each dimension. Following Figure 2.15 shows a sample lexicon entry for two dimensions viz. SYN (syntactic) and SEM (semantic).

Figure 2.15 Lexical Entry for ‘wants’
XDG parser is capable of processing these dimensions simultaneously.

### 2.4.2.11 Computational Paninian Grammar (CPG)

CPG (Bharati, Chaitanya, & Sangal, 1995) is based on the legendary master work by ancient Indian linguist Panini (500 B.C.). Panini wrote a theory about how to analyze sentences in Natural Languages with special reference to extant Sanskrit Language. As Sanskrit is supposedly mother language for most of the Indian Languages. Paninian tradition is unanimously the oldest reported tradition of linguistics in the history of mankind. It is based on the modern concept of dependency or modifier-modified paradigm. Panini’s work is divided into 08 chapters of 04 sections each (Kulkarni, 2008). Asthadhyayi, Shiva Sutram, Dhatupada, and Ganpada are main highlights of Panini’s theory. Ashtadhyayi has around 4000 rules, wrote in very concise and compressed way. He has discussed 2000 verbal roots (Dhatupada) and 261 lexical items (Ganapada). Panini’s theory is known as ‘Karaka Theory’. Roughly speaking, a sentence at core level is nothing but sequence of nouns and verb(s) and every sentence is anchored at the verb. All nouns in the sentence are related to verb with special semantic relation called ‘karaka’. One can think of these karaka roles as similar but not exactly same as theta roles in western linguistics. Since Indian Languages are morphologically very rich, karaka theory is syntactico-semantic. One can identify a particular karaka from syntactic cues provided by post position markers, more scientifically called as ‘Vibhakti symbols’ in Indian Linguistic tradition.

### 2.5 Findings of the Literature Survey

As stated in the preamble of this chapter, this survey exercise was very much fruitful as anticipated towards, designing and development of HinMaT framework. It gave us complete exposure of the MT field and state-of-the-art. Followings are the findings of this exercise.

1. Literature survey helped in giving clear understanding of different approaches and architectures/paradigms for MT and their inter relationships. Considering the divergence issues in Hindi and Marathi and strengths and weaknesses of various MT paradigms, we have decided to use Transfer architecture and Shake-&-Bake MT paradigm for HinMaT.
2. The MT initiatives revealed that good amount of MT systems have been developed for European Languages. Most of the systems developed using rule based MT paradigm are based on syntactic transfer in Vauquois pyramid.

3. SMT paradigm is lucrative option for the languages for whom good amount of online resources are available. SMT is ‘hot cake’ option these days amongst the research community. But it is constrained by large data size.

4. Enough initiatives have been taken in Indian context. As such there is more demand for English to Indian languages MT systems, more projects have been done considering English as SL. Most of the Indian MT projects are funded by Govt. agencies like TDIL (Technology Development for Indian Languages), DIT(Dept. of Information Technology), MoCIT, New Delhi, UGC(University Grants Commission), New Delhi and DST(Department of Science and Technology), New Delhi etc.

5. It is important to note here that as such no Hindi-Marathi MT system is available in RBMT paradigm. However one online system available i.e. Google Translate is presently available. It is based on SMT paradigm and quality of output is generally rated as poor. This fact underlines the importance and uniqueness of proposed research work.

6. Comparative study of constituency and dependency school revealed that dependency grammar formalisms are better for modelling free-word-order languages like Indian Languages.

2.6 Summary

This chapter discusses the history of MT field in lucid way followed by MT approaches and various development paradigms. Various historical MT systems are enumerated in support of the MT history. MT initiatives in Indian context are described in detail by summarizing various Indian MT systems, classified on the basis of source language. The chapter also surveys important grammar formalisms from constituency and dependency school. These include CFG, LFG, PSG, FUG, HDPSG, FDG, DUG, XDG and Computational Paninian Grammar (CPG) etc. The findings of
the literature survey exercise reports uniqueness of the work as well as strengths, weaknesses of grammar formalisms under two prevailing schools.