B.Sc. in Computer Science and Engineering Thesis

Bangla To English Machine Translation

Submitted by

Nazifa Azam Khan
200914037

A.S.M.Muntaheen
200914052

Ankur Bhattacharjee
200914017

Supervised by

Dr. Mohammad Nurul Huda
Associate Professor
Dept. of CSE (United International University)

Department of Computer Science and Engineering
Military Institute of Science and Technology

December 2012

CERTIFICATION

This thesis paper titled **“Bangla to English Machine Trnaslation”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering on December 2012.

Group Members:

Nazifa Azam Khan
A.S.M.Muntaheen
Ankur Bhattacharjee

Supervisor:

_____-

Dr. Mohammad Nurul Huda
Associate Professor and MSCSE Coordinator
Dept. of CSE (United International University)
Address:
UIU Bhaban,House-80,
Road-8/A(Old-15),
Satmasjid Road,Dhanmondi,
Dhaka-1209,Bangladesh.

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper is the outcome of the investigation and research carried out by the following students under the supervision of Dr. Mohammad Nurul Huda, Associate Professor and MSCSE Coordinator, Dept. of CSE, United International University, UIU Bhaban, House-80, Road -8/A, Satmasjid Road, Dhanmondi, Dhaka-1209, Bangladesh.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

_____-

Nazifa Azam Khan
200914037

_____-

A.S.M.Muntaheen
200914052

_____-

Ankur Bhattacharjee
200914017

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor Dr. Mohammad Nurul Huda, Associate Professor and MSCSE Coordinator, Dept. of CSE, (United International University), UIU Bhaban, House-80, Road-8/A, Satmasjid Road, Dhanmondi, Dhaka-1209, Bangladesh, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2012

Nazifa Azam Khan
A.S.M.Muntaheen
Ankur Bhattacharjee

ABSTRACT

Machine Translation (MT) refers to the use of computers for the task of translating automatically from one language to another. The differences between languages and especially the inherent ambiguity of language make MT a very difficult problem. Traditional approaches to MT have relied on humans supplying linguistic knowledge in the form of rules to transform text in one language to another. Given the vastness of language, this is a highly knowledge intensive task. Statistical MT is a radically different approach that automatically acquires knowledge from large amounts of training data. This knowledge, which is typically in the form of probabilities of various language features, is used to guide the translation process. This report provides an overview of MT techniques, and looks in detail at the basic statistical model.

Keywords: Machine Translation, Statistical Machine Translation, Corpus Based Approach, Transfer Based Approach, Target Language, Source Language.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

MT : Machine Translation

SL : Source Language

TL : Target Language

CBMT : Corpus Based Machine Translation

TBMT : Transfer Based Machine Translation

SMT : Statistical Machine Translation

DMT : Direct Machine Translation

IMT : Interlingua Machine Translation

S-V-O: Subject- verb- object

S-O-V: Subject-object-verb

CFG : Context Free Grammar

LIST OF SYMBOLS

p	: Probability Notation
Γ	: Expression of Loci for sequence fair
$\lceil logn \rceil$	: Ceiling of log n
n !	: Permutation of n
∑	: Summation Notation
β	: Clonal Factor
Π	: Notation of Product

CHAPTER 1
INTRODUCTION

1.1 Machine Translation: an Overview

The mechanization of translation has been one of humanity's oldest dreams. In the twentieth century it has become a reality, in the form of computer programs capable of translating a wide variety of texts from one natural language into another. But, as ever, reality is not perfect. There are no 'translating machines' which, at the touch of a few buttons, can take any text in any language and produce a perfect translation in any other language without human intervention or assistance. That is an ideal for the distant future, if it is even achievable in principle, which many doubt.

What has been achieved is the development of programs which can produce 'raw' translations of texts in relatively well-defined subject domains, which can be revised to give good-quality translated texts at an economically viable rate or which in their unedited state can be read and understood by specialists in the subject for information purposes. In some cases, with appropriate controls on the language of the input texts, translations can be produced automatically that are of higher quality needing little or no revision.

These are solid achievements by what is now traditionally called Machine Translation (henceforth in this book, MT), but they have often been obscured and misunderstood. The public perception of MT is distorted by two extreme positions. On the one hand, there are those who are unconvinced that there is anything difficult about analyzing language, since even young children are able to learn languages so easily; and who are convinced that anyone who knows a foreign language must be able to translate with ease. Hence, they are unable to appreciate the difficulties of the task or how much has been achieved. On the other hand, there are those who believe that because automatic translation of Shakespeare, Goethe, Tolstoy and lesser literary authors is not feasible there is no role for any kind of computer-based

translation. They are unable to evaluate the contribution which less than perfect translation could make either in their own work or in the general improvement of international communication. Machine translation can be defined as the use of computers to automate some or all of the process of translating from one language to another. ***Machine Translation uses***

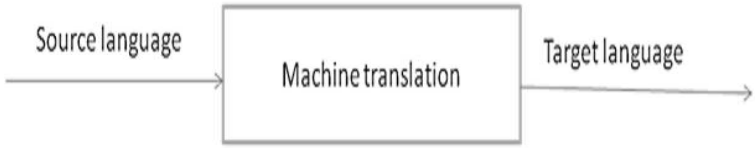


Figure 1.1: Machine Translation

ideas and techniques:

- Linguistics
- Computer science
- Artificial intelligence
- Translation theory
- Statistics

Commercially interesting:

- US has invested in machine translation for intelligence purposes.
- EU spends more than one billion on translation costs each year.
- Machine translation is used in Universal Network Language.
- Machine translation is used to translate technical documents , reports , instruction manuals etc.

1.2 Some preliminary definitions

The term Machine Translation (MT) is the now traditional and standard name for computerized systems responsible for the production of translations from one natural language into another, with or without human assistance. Earlier names such as 'mechanical translation' and 'automatic translation' are now rarely used in English; but their equivalents in other languages are still common (e.g. French traduction automatique , Russian avtomati?eskii perevod). The term does not include computer-based translation tools which support translators by providing access to dictionaries and remote terminology databases, facilitating the transmission and reception of machine-readable texts, or interacting with word processing, text editing or printing equipment. It does, however, include systems in which translators or other users assist computers in the production of translations, including various combinations of text preparation, on-line interactions and subsequent revisions of output. The boundaries between Machine-Aided Human Translation (MAHT) and Human-Aided Machine Translation (HAMT) are often uncertain and the term Computer-Aided (or Computer-Assisted) Translation (both CAT) can sometimes cover both. But the central core of MT itself is the automation of the full translation process. Although the ideal may be to produce high-quality translations, in practice the output of most MT systems is revised (post-edited). In this respect, MT output is treated no differently than the output of most human translators which is normally revised by another translator before dissemination. However, the types of errors produced by MT systems do differ from those of human translators. While post editing is the norm, there are certain circumstances when MT output may be left unedited (as a raw translation) or only lightly corrected, e.g. if it is intended only for specialists familiar with the subject of the text. Output may also serve as a rough draft for a human translator, as a pre-translation.

The translation quality of MT systems may be improved - not only, of course, by developing better methods - by imposing certain restrictions on the input. The system may be designed, for example, to deal with texts limited to the sublanguage (vocabulary and grammar) of a particular subject field (e.g. polymer chemistry) and/or document type (e.g. patents). Alternatively, input texts may be written in a controlled language, which reduces potential ambiguities and restricts the complexity of sentence structures. This option is often referred to as pre-editing, but the term can also be used for the marking of input texts to indicate

proper names, word divisions, prefixes, suffixes, phrase boundaries, etc. Finally the system itself may refer problems of ambiguity and selection to human operators (usually translators, though some systems are designed for use by the original authors) for resolution during the processes of translation itself, i.e. in an interactive mode. Systems are designed either for one particular pair of languages (bilingual systems) or for more than two languages (multilingual systems), either in one direction only (uni-directional systems) or in both directions (bi-directional systems). In overall system design, there are three basic types. The first (and also historically oldest) is generally referred to as the direct translation approach: the MT system is designed in all details specifically for one particular pair of languages in one direction, e.g. Russian as the language of the original texts, the source language, and English as the language of the translated texts, the target language. Source texts are analysed no more than necessary for generating texts in the other language. The second basic type is the Interlingua approach, which assumes the possibility of converting texts to and from 'meaning' representations common to more than one language. Translation is thus in two stages: from the source language to the Interlingua, and from the interlingua into the target language. Programs for analysis are independent from programs for generation; in a multilingual configuration, any analysis program can be linked to any generation program. The third type is the less ambitious transfer approach. Rather than operating in two stages through a single interlingual meaning representation, there are three stages involving, usually, syntactic representations for both source and target texts. The first stage converts texts into intermediate representations in which ambiguities have been resolved irrespective of any other language. In the second stage these are converted into equivalent representations of the target language; and in the third stage, the final target texts are generated. Analysis and generation programs are specific for particular languages and independent of each other. Differences between languages, in vocabulary and structure, are handled in the intermediary transfer program. Within the stages of analysis and generation, most MT system exhibit clearly separated components dealing with different levels of linguistic description: morphology, syntax, semantics. Hence, analysis may be divided into morphological analysis (e.g. identification of word endings), syntactic analysis (identification of phrase structures, etc.) and semantic analysis (resolution of lexical and structural ambiguities). Likewise, generation (or synthesis) may pass through levels of semantic, syntactic and morphological generation. In transfer systems, there may be separate components dealing with lexical transfer (selection

of vocabulary equivalents) and structural transfer (transformation of source text structures into equivalent target text ones).In many older systems (particularly those of the direct translation type), rules for analysis, transfer and generation were not always clearly separated. Some also mixed linguistic data (dictionaries and grammars) and computer processing rules and routines. Later systems exhibit various degrees of modularity, so that system components, data and programs can be adapted and changed independently of each other.

1.3 The aims of Machine Translation

Most translation in the world is not of texts which have high literary and cultural status. The great majority of professional translators are employed to satisfy the huge and growing demand for translations of scientific and technical documents, commercial and business transactions, administrative memoranda, legal documentation, instruction manuals, agricultural and medical text books, industrial patents, publicity leaflets, newspaper reports, etc. Some of this work is challenging and difficult. But much of it is tedious and repetitive, while at the same time requiring accuracy and consistency. The demand for such translations is increasing at a rate far beyond the capacity of the translation profession. The assistance of a computer has clear and immediate attractions. The practical usefulness of an MT system is determined ultimately by the quality of its output. But what counts as a 'good' translation, whether produced by human or machine, is an extremely difficult concept to define precisely. Much depends on the particular circumstances in which it is made and the particular recipient for whom it is intended. Fidelity, accuracy, intelligibility, appropriate style and register are all criteria which can be applied, but they remain subjective judgments. What matters in practice, as far as MT is concerned, is how much has to be changed in order to bring output up to a standard acceptable to a human translator or reader. With such a slippery concept as translation, researchers and developers of MT systems can ultimately aspire only to producing translations which are 'useful' in particular situations - which obliges them to define clear research objectives - or, alternatively, they seek suitable applications of the 'translations' which in fact they are able to produce. Nevertheless, there remains the higher ideal of equaling the best human translation. MT is part of a wider sphere of 'pure research' in computer based natural language processing in Computational Linguistics and Artificial Intelligence, which explore the basic mechanisms of language and mind by model-

ing and simulation in computer programs. Research on MT is closely related to these efforts, adopting and applying both theoretical perspectives and operational techniques to translation processes, and in turn offering insights and solutions from its particular problems. In addition, MT can provide a 'test-bed' on a larger scale for theories and techniques developed by small-scale experiments in computational linguistics and artificial intelligence. In brief, MT is not in itself an independent field of 'pure' research. It takes from linguistics, computer science, artificial intelligence, translation theory, any ideas, methods and techniques which may serve the development of improved systems. It is essentially 'applied' research, but a field which nevertheless has built up a substantial body of techniques and concepts which can, in turn, be applied in other areas of computer-based language processing.

1.4 Applications of Machine Translation

While no system provides the holy grail of fully automatic high-quality machine translation of unrestricted text, many fully automated systems produce reasonable output. The quality of machine translation is substantially improved if the domain is restricted and controlled. Despite their inherent limitations, MT programs are used around the world. Probably the largest institutional user is the European Commission. The MOLTO project, for example, coordinated by the University of Gothenburg, received more than 2.375 million euros project support from the EU to create a reliable translation tool that covers a majority of the EU languages. Google has claimed that promising results were obtained using a proprietary statistical machine translation engine. The statistical translation engine used in the Google language tools for Arabic to English and Chinese to English had an overall score of 0.4281 over the runner-up IBM's BLEU-4 score of 0.3954 (Summer 2006) in tests conducted by the National Institute for Standards and Technology. With the recent focus on terrorism, the military sources in the United States have been investing significant amounts of money in natural language engineering. In-Q-Tel (a venture capital fund, largely funded by the US Intelligence Community, to stimulate new technologies through private sector entrepreneurs) brought up companies like Language Weaver. Currently the military community is interested in translation and processing of languages like Arabic, Pashto, and Dari. The Information Processing Technology Office in DARPA hosts programs like TIDES and Babylon Translator. US Air Force has awarded a 1 million contract to develop a language translation

technology. The notable rise of social networking on the web in recent years has created yet another niche for the application of machine translation software - in utilities such as Facebook, or instant messaging clients such as Skype, GoogleTalk, MSN Messenger, etc. - allowing users speaking different languages to communicate with each other. Machine translation applications have also been released for most mobile devices, including mobile telephones, pocket PCs, PDAs, etc. Due to their portability, such instruments have come to be designated as mobile translation tools enabling mobile business networking between partners speaking different languages, or facilitating both foreign language learning and unaccompanied traveling to foreign countries without the need of the intermediation of a human translator.

1.5 Translation Process

Even though no one is capable of providing an exact list of rules that would allow to arrive at a perfect translation, there are some procedures and methods, knowledge of which may facilitate translators' work. In order to have an idea about translation itself and be able to produce texts in various languages, one should get familiar with the process and theory of translation. The awareness of both notions may provide necessary advice and clues. What is more, it may be beneficiary for the translators' competence: increasing the quality of their work; enabling them to deliver the translation according to the rules, style, and grammar of the TL; allowing for quick, accurate, clear and naturally sounding translation. Every translator adapts their own approach towards the process of translation, nevertheless it always involves working in subsequent steps. The following passage describes two different models of translation process: the two-phase model and the three-phase model that may help to arrange the act of a text production. Adapting of the first model includes working in two sequential phases, namely analysis (decoding) and synthesis (recoding), whereas the second model adaptation additionally incorporates transfer (transcoding) phase.

According to Nord (2005), the first step - analysis, includes dissolution of grammatical, semantic and stylistic elements which is to help a translator handle the meaning (both explicit and implicit). In the second step, a translator is supposed to choose his or her strategy, decide whether the text function is to be changed or preserved. Whereas in the last step, the final product - a target text, conforming to the needs of the TT receivers is produced.

In order to be more competent, besides being acknowledged with the phases of the translation process, a professional translator should also be aware of the theory of translation including translation strategies, procedures and methods. Translation strategy may be defined as a plan undertaken by a translator to achieve a certain translation goal. The term strategy incorporates techniques, methods as well as procedures. Newmark (1988) mentions the difference between translation methods and translation procedures. He writes that,while translation methods relate to whole texts, translation procedures are used for sentences and the smaller units of language.It should also be stressed that a strategy, besides concerning the whole text, is undertaken on the basis on the initiator's needs, text type and a purpose that it is to serve. Procedure, on the other hand, is a more narrow notion, applied to solve a specific problem by turning to a dictionary or asking other translators for help.

1.6 Obstacles in Machine Translation

The major obstacles to translating by computer are, as they have always been, not computational but linguistic. They are the problems of lexical ambiguity, of syntactic complexity, of vocabulary differences between languages, of elliptical and 'ungrammatical' constructions, of, in brief, extracting the 'meaning' of sentences and texts from analysis of written signs and producing sentences and texts in another set of linguistic symbols with an equivalent meaning. Consequently, MT should expect to rely heavily on advances in linguistic research, particularly those branches exhibiting high degrees of formalization, and indeed it has and will continue to do so. But MT cannot apply linguistic theories directly: linguists are concerned with explanations of the underlying 'mechanisms' of language production and comprehension, they concentrate on crucial features and do not attempt to describe or explain everything. MT systems, by contrast, must deal with actual texts. They must confront the full range of linguistic phenomena, the complexities of terminology, misspellings, neologisms, aspects of 'performance' which are not always the concern of abstract theoretical linguistics.

1.6.1 Ambiguity

Words and phrases in one language often map to multiple words in another language. For example, in the sentence, **I went to the bank,** it is not clear whether the *mound of sand* (nodir tir in Bangla) sense or the *financial institution* (bank) sense is being used. This will usually be clear from the context, but this kind of disambiguation is generally non-trivial [Nancy and Veronis, 1998]. Also, each language has its own idiomatic usages which are difficult to identify from a sentence. For example, **India and Pakistan have broken the ice finally.** Phrasal verbs are another feature that are difficult to handle during translation. Consider the use of the phrasal verb bring up in the following sentences, *They brought up the child in luxury.* (lalon palon) *They brought up the table to the first floor.* (upore tola) *They brought up the issue in the house.* (bishoi utthapon kora) Yet another kind of ambiguity that is possible is structural ambiguity: **Flying planes can be dangerous.** This can be translated in Bangla as either of the following two sentences. *urojahaj urano bipodjonok hote pare uronto urojahaj bipodjonok hote pare* depending on whether it is the planes that are dangerous or the occupation of flying them that is dangerous!

1.6.2 Structural Differences

Just as English follows a Subject-Verb-Object (SVO) ordering in sentences, each language follows a certain sentence structure. Bangla, for example, is a Subject- Object-Verb language. Apart from this basic feature, languages also differ in the structural (or syntactic) constructions that they allow and disallow. These differences have to be respected during translation. For instance, post-modifiers in English become pre-modifiers in Bangla, as canbe seen from the following pair of sentences. These sentences also illustrate the SVO and SOV sentence structure in these languages. Here, S is the subject of the sentence, S_m is the subject modifier, and similarly for the verb (V) and the object (O).

The president of America will visit the capital of Bangladesh.

(S) (S_m) (V) (O) (O_m)

americar rastropoti bangladesher rajdhani sofor korben

(S_m) (S) (O_m) (O) (V)

1.6.3 Vocabulary Differences

Languages differ in the way they lexically divide the conceptual space, and sometimes no direct equivalent can be found for a particular word or phrase of one language in another. Consider the sentence,

tendulkarer beter kanai bol legechilo

kanai as a verb has no equivalent in English, and this sentence has to be translated as,

Tendulkar has edged the ball.

See [Hutchins and Somers, 1992] for more examples of vocabulary differences between languages and also other problems in MT.

CHAPTER 2
BACKGROUND

2.1 Brief history of MT

The use of mechanical dictionaries to overcome barriers of language was first suggested in the 17th century. Both Descartes and Leibniz speculated on the creation of dictionaries based on universal numerical codes. Actual examples were published in the middle of the century by Cave Beck, Athanasius Kircher and Johann Becher. The inspiration was the 'universal language' movement, the idea of creating an unambiguous language based on logical principles and iconic symbols (as the Chinese characters were believed to be), with which all humanity could communicate without fear of misunderstanding. Most familiar is the interlingua elaborated by John Wilkins in his 'Essay towards a Real Character and a Philosophical Language' (1668). In subsequent centuries there were many more proposals for international languages (with Esperanto as the best known), but few attempts to mechanize translation until the middle of this century. In 1933 two patents appeared independently in France and Russia. A French-Armenian, George Artsrouni, had designed a storage device on paper tape which could be used to find the equivalent of any word in another language; a prototype was apparently demonstrated in 1937. The proposal by the Russian, Petr Smirnov-Troyanskii, was in retrospect more significant. He envisaged three stages of mechanical translation: first, an editor knowing only the source language was to undertake the 'logical' analysis of words into their base forms and syntactic functions; secondly, a machine was to transform sequences of base forms and functions into equivalent sequences in the target language; finally, another editor knowing only the target language was to convert this output into the normal forms of that language. Although his patent referred only to the machine which would undertake the second stage, Troyanskii believed that "the process of logical analysis could itself be mechanised". Troyanskii was ahead of his time and was unknown outside Russia when, within a few years of their invention, the possibility of using

computers for translation was first discussed by Warren Weaver of the Rockefeller Foundation and Andrew D. Booth, a British crystallographer. On his return to Birkbeck College (London) Booth explored the mechanization of a bilingual dictionary and began collaboration with Richard H. Richens (Cambridge), who had independently been using punched cards to produce crude word-for-word translations of scientific abstracts. However, it was a memorandum from Weaver in July 1949 which brought the idea of MT to general notice and suggested methods: the use of wartime cryptography techniques, statistical analysis, Shannon's information theory, and exploration of the underlying logic and universal features of language. Within a few years research had begun at a number of US centres, and in 1951 the first full-time researcher in MT was appointed: Yehoshua Bar-Hillel at MIT. A year later he convened the first MT conference, where the outlines of future research were already becoming clear. There were proposals for dealing with syntax, suggestions that texts should be written in controlled languages, arguments for the construction of sublanguage systems, and recognition of the need for human assistance (preand post-editing) until fully automatic translation could be achieved. For some, the first requirement was to demonstrate the technical feasibility of MT. Accordingly, at Georgetown University Leon Dostert collaborated with IBM on a project which resulted in the first public demonstration of a MT system in January 1954. A carefully selected sample of Russian sentences was translated into English, using a very restricted vocabulary of 250 words and just six grammar rules. Although it had little scientific value, it was sufficiently impressive to stimulate the large-scale funding of MT research in the United States and to inspire the initiation of MT projects elsewhere in the world, notably in the Soviet Union. For the next decade many groups were active: some adopting empirical trial-and-error approaches, often statistics-based, with immediate working systems as the goal; others took theoretical approaches, involving fundamental linguistic research, aiming for long-term solutions. The contrasting methods were usually described at the time as 'brute-force' and 'perfectionist' respectively. Examples of the former were the lexicographic approach at the University of Washington(Seattle), later continued by IBM in a Russian-English system completed for the US Air Force, the statistical 'engineering' approach at the RAND Corporation, and the methods adopted at the Institute of Precision Mechanics in the Soviet Union, and the National Physical Laboratory in Great Britain. Largest of all was the group at Georgetown University, whose successful Russian-English system is now regarded as typical of this 'first generation' of MT research. Centres

of theoretical research were at MIT, Harvard University, the University of Texas, the University of California at Berkeley, at the Institute of Linguistics in Moscow and the University of Leningrad, at the Cambridge Language Research Unit (CLRU), and at the universities of Milan and Grenoble. In contrast to the more pragmatically oriented groups where the 'direct translation' approach was the norm, some of the theoretical projects experimented with early versions of interlingua and transfer systems (e.g. CLRU and MIT, respectively).

Much of the research of this period was of lasting importance, not only for MT but also for computational linguistics and artificial intelligence - in particular, the development of automated dictionaries and of techniques for syntactic analysis - and many theoretical groups made significant contributions to linguistic theory. However, the basic objective of building systems capable of producing good translations was not achieved. Optimism had been high, there were many predictions of imminent breakthroughs, but disillusionment grew as the complexity of the linguistic problems became more and more apparent. In a 1960 review of MT progress, Bar-Hillel criticized the prevailing assumption that the goal of MT research should be the creation of fully automatic high quality translation (FAHQT) systems producing results indistinguishable from those of human translators. He argued that the 'semantic barriers' to MT couldin principle only be overcome by the inclusion of vast amounts of encyclopedic knowledge about the 'real world'. His recommendation was that MT should adopt less ambitious goals, it should build systems which made cost-effective use of human-machine interaction. In 1964 the government sponsors of MT in the United States formed the Automatic Language Processing Advisory Committee (ALPAC) to examine the prospects. In its influential 1966 report it concluded that MT was slower, less accurate and twice as expensive as human translation and stated that "there is no immediate or predictable prospect of useful Machine Translation". It saw no need for further investment in MT research; instead it recommended the development of machine aids for translators, such as automatic dictionaries, and continued support of basic research in computational linguistics. The ALPAC report was widely condemned as narrow, biased and shortsighted - it was certainly wrong to criticize MT because output had to be post-edited, and it misjudged the economic factors- but large-scale financial support of current approaches could not continue. Its influence was profound, bringing 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. In the

following decade MT research took place largely outside the United States, in Canada and in Western Europe, and virtually ignored by the scientific community. American activity had concentrated on English translations of Russian scientific and technical materials. In Canada and Europe the needs were quite different: the Canadian bicultural policy created a demand for English-French (and to a less extent French-English) translation beyond the capacity of the market, and the European Economic Community (as it was then known) was demanding translations of scientific, technical, administrative and legal documentation from and into all the Community languages.

A research group was established at Montreal which, though ultimately unsuccessful in building a large English-French system for translating aircraft manuals, is now renowned for the creation in 1976 of the archetypal 'sublanguage' system Mto for translating weather reports for daily public broadcasting. In 1976 the Commission of the European Communities decided to install an English-French system called Systran, which had previously been developed by Peter Toma (once a member of the Georgetown team) for Russian-English translation for the US Air Force, and had been in operation since 1970. In subsequent years, further systems for French-English, English-Italian, English-German and other pairs have been developed for the Commission. In the late 1970s, it was also decided to fund an ambitious research project to develop a multilingual system for all the Community languages, based on the latest advances in MT and in computational linguistics. This is the Eurotra project, which involves research groups in all member states. For its basic design, Eurotra owes much to research at Grenoble and at Saarbrcken. During the 1960s the French group had built an 'interlingua' system for Russian-French translation (not purely interlingual as lexical transfer was still bilingual); however, the results were disappointing and in the 1970s it began to develop the influential transfer-based Ariane system. The Saarbrcken group had also been building its multilingual 'transfer' system SUSY since the late 1960s. It was now the general consensus in the MT research community that the best prospects for significant advances lay in the development of transfer-based systems. The researchers at the Linguistics Research Center (LRC) at Austin, Texas (one of the few to continue after ALPAC) had come to similar conclusions after experimenting with an interlingua system and was now developing its transfer-based METAL system; and in Japan work had begun at Kyoto University on the Mu transfer system for Japanese-English translation. The

Eurotra group adopted the same basic approach, although it found subsequently that the demands of large-scale multilinguality led to the incorporation of many interlingual features. However, during the 1980s the transfer-based design has been joined by new approaches to the interlingua idea. Most prominent is the research on knowledgebased systems, notably at Carnegie Mellon University, Pittsburgh, which are founded on developments of natural language understanding systems within the Artificial Intelligence (AI) community. The argument is that MT must go beyond purely linguistic information (syntax and semantics); translation involves 'understanding' the content of texts and must refer to knowledge of the 'real world'. Such an approach implies translation via intermediate representations based on (extra-linguistic) 'universal' elements. Essentially non-AI-oriented interlingua approaches have also appeared in two Dutch projects: the DLT system at Utrecht based on a modification of Esperanto and the Rosetta system at Phillips (Eindhoven) which is experimenting with Montague semantics as the basis for an interlingua More recently, yet other alternatives have emerged. For many years, automatic translation of speech was considered Utopian, but advances in speech recognition and speech production have encouraged the foundation of projects in Great Britain (British Telecom) and in Japan (Advanced Telecommunications Research, ATR). The sophistication of the statistical techniques developed by speech research has revived interest in the application of such methods in MT systems; the principal group at present is at the IBM laboratories at Yorktown Heights, NY The most significant development of the last decade, however, is the appearance of commercial MT systems. The American products from ALPSystems, Weidner and Logos were joined by many Japanese systems from computer companies (Fujitsu, Hitachi, Mitsubishi, NEC, Oki, Sanyo, Sharp, Toshiba), and in the later 1980s by Globalink, PC-Translator, Tovna and the METAL system developed by Siemens from earlier research at Austin, Texas. Many of these systems, particularly those for microcomputers, are fairly crude in the linguistic quality of their output but are capable of cost-effective operation in appropriate circumstances . As well as these commercial systems, there have been a number of in-house systems, e.g. the Spanish and English systemsdeveloped at the Pan-American Health Organization (Washington, DC), and the systems designed by the Smart Corporation for Citicorp, Ford, and the Canadian Department of Employment and Immigration. Many of the Systran installations are tailor-made for particular organisations (Arospatiale, Dornier, NATO, General Motors).

Nearly all these operational systems depend heavily on post-editing to produce acceptable translations. But pre-editing is also widespread: in some systems, for instance, operators are required, when inputting text, to mark word boundaries or even indicate the scope of phrases and clauses. At Xerox, texts for translation by Systran are composed in a controlled English vocabulary and syntax; and a major feature of the Smart systems is the pre-translation editor of English input.The revival of MT research in the 1980s and the emergence of MT systems in the marketplace have led to growing public awareness of the importance of translation tools. There may still be many misconceptions about what has been achieved and what may be possible in the future , but the healthy state of MT is reflected in the multiplicity of system types and of research designs which are now being explored , many undreamt of when MT was first proposed in the 1940s. Further advances in computer technology, in Artificial Intelligence and in theoretical linguistics suggest possible future lines of investigation, while different MT user profiles (e.g. the writer who wants to compose a text in an unknown language) lead to new designs. But the most fundamental problems of computer-based translation are concerned not with technology but with language, meaning, understanding, and the social and cultural differences of human communication.

2.2 Generations and Types of Machine Translation

Machine translation systems can be divided in two generations direct systems and indirect systems. First generation systems are known as direct systems. In such systems, translation is done word by word or phrase by phrase. In such systems very minimal linguistic analysis of input text is conducted (Hutchins and Somers 1992). This architecture is still being extensively used in commercial MT systems. The main idea behind direct systems is to analyze the input text to the extent that some transformational rules can be applied. This analysis could be parts of speech of words or some phrasal level information. Then using a bilingual dictionary, source language words are replaced with target language words and some rearrangement rules are used to modify the word order according to the target language (Arnold et al. 1993).

This architecture is very robust because it does not fail on any erroneous or ungrammatical input. Since the analysis level is very shallow and the system contains very limited gram-

matical information, it hardly considers anything ungrammatical. In the worst case if the rule does not apply to the input, the input is passed on without any alteration as output. This kind of system is hard to extend because all the rules are written in one direction and are language specific. To make another language pair work, all the rules have to be rewritten. Since the system does not perform very deep analysis, its time complexity is low. These systems work very well for closely related languages but are not suitable for modeling languages with diverse syntactic nature. Since the system does not explicitly contain the grammatical rules of the target language, there is a chance that the output will not be grammatical but it will be similar to the target language (Arnold et al. 1993)

Owing to the fact that linguistic information helps an MT system to produce better quality target language translation, with the advance of computing technology, MT researchers started to develop methods to capture and process the linguistics of sentences. This was when the era of second generation MT systems started. Second generation machine translation systems are called indirect systems. In such systems the source language structure is analyzed and text is transformed into a logical form. The target language translation is then generated from the logical form of the text (Hutchins and Somers 1992).The transition from direct systems to indirect systems is illustrated in Figure 2.1, taken from (Hutchins and Somers 1992, pg. 107).

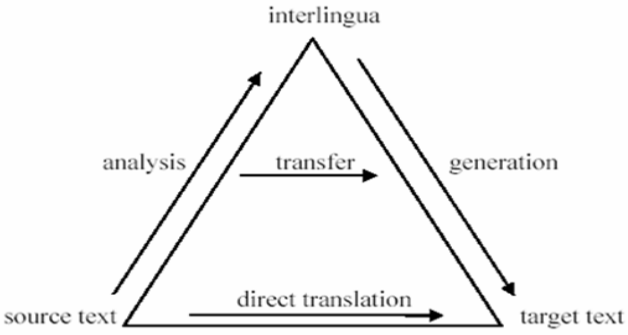


Figure 2.1: Transfer and interlingua 'pymarid' diagram

CHAPTER 3
APPROACHES OF MACHINE TRANSLATION

Machine translation can use a method based on linguistic rules, which means that words will be translated in a linguistic way - the most suitable (orally speaking) words of the target language will replace the ones in the source language.It is often argued that the success of machine translation requires the problem of natural language understanding to be solved first.

Generally, rule-based methods parse a text, usually creating an intermediary, symbolic representation, from which the text in the target language is generated. According to the nature of the intermediary representation, an approach is described as interlingua machine translation or transfer-based machine translation. These methods require extensive lexicons with morphological, syntactic, and semantic information, and large sets of rules.Given enough data, machine translation programs often work well enough for a native speaker of one language to get the approximate meaning of what is written by the other native speaker. The difficulty is getting enough data of the right kind to support the particular method. For example, the large multilingual corpus of data needed for statistical methods to work is not necessary for the grammar-based methods. But then, the grammar methods need a skilled linguist to carefully design the grammar that they use. To translate between closely related languages, a technique referred to as shallow-transfer machine translation may be used.

3.1 Word for word approach

A common misconception among students, specially those not familiarized with machine translation (MT), is that MT systems follow a strategy similar to that implemented in early MT programs in the 50's. This strategy, usually known as word-for-word translation1, ignores inter-word dependencies considering each word in a sentence in isolation, and lacks any kinds of intermediate representations. Obviously, this kind of strategies produce very poor results, even when the source language (SL) and the target language (TL) share similar lexical, morphological, syntactical and semantical structures. In fact, this basic approach to MT is what we might expect if we asked a non-expert to design a MT system. The outcome would be comparable to that obtained from "someone with a very cheap bilingual dictionary and only the most rudimentary knowledge of the grammar of the target language: frequent mistranslations at the lexical level and largely inappropiate syntax structures which mirrored too closely those of the source language" (Hutchins and Somers 1992, p. 72). On the one hand, current real MT programs implement techniques much more advanced than word-for-word translation. Although there are a lot of situations in which they still keep on generating wrong translations, MT systems perform a deep analysis on sentence as a whole, implementing processes such as context-dependent homograph2 resolution, special processing of multiword units (such as idioms), word reordering, agreement enforcement, or exception handling.

Nowadays, on the other hand, commercial systems whose translations may be considered acceptable to some level are available at low or medium prices, or even freely on the Internet; they have become an affordable tool for helping the task of the machine translation instructor. Our proposal is a laboratory assignment where students discover some of the multiple processes which go beyond a simple word-for-word strategy and are implemented in real MT systems, and how they are better than the word-for-word approach. Laboratory work is mainly designed for non-computer-science majors but it may be used as well with computer-science majors. The source language (SL) is English and the target language (TL) is Spanish. It has been successfully tested for six years with third-year translation majors with very basic computer skills in general. Machine translation majors learn also the advantages and disadvantages of using MT programs: these programs are enormously imperfect but they still can be useful. Further more, the assignment may help non-computer students

to give up some misconceptions (sometimes a complete ignorance) about the algorithmic behavior of computers.

3.1.1 Algorithm

A word-for-word translation strategy can be described as a three-phase process (Hutchins and Somers 1992, p. 72):

a) The first phase consists of a rudimentary morphological analysis where each superficial form (SF) in the SL is converted into its corresponding lexical form (LF). Homograph disambiguation is not implemented in this approach.

b) A bilingual dictionary is looked up in the second phase in order to translate each LF to its corresponding LF in the TL.

c) Finally, the LF in the TL is inflected to obtain the translation (some local reorderings are probably done in this phase as well).

3.1.2 Examples

Word for word translation approach uses a machine-readable bilingual dictionary to translate each word in a text.

Table 3.1 Database of word for word

Bangla	**English**
Ami	I
Vat	Rice
Khai	Eat
.	.
.	.
.	.

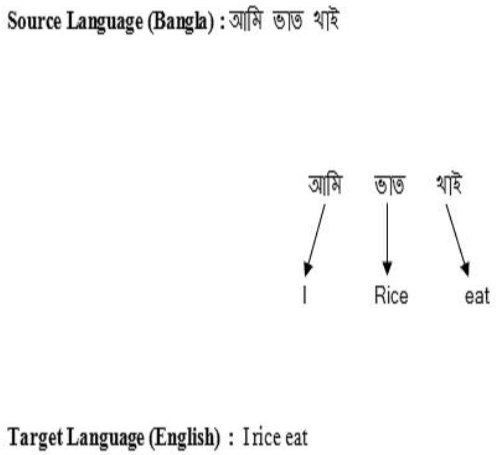


Figure 3.1: Example of word for word approach

3.1.3 Difficulties

Though word for word translation is easy to implement and its results give a rough idea about what the text is about it has some difficulties. It problems with word order means that this results in low quality translation.

3.2 Corpus based approach

The information revolution and technological innovations have driven the development of language industries and the expansion of multilingualism. The use of machine translation has experienced unprecedented growth with many diverse new techniques and demands. However, the prime objective of researchers and businessmen, in an Internet-dominated environment, has been the rapid development of translation systems that are both accurate and effective.

This technological development, along with the huge volume of translations available in different languages, point toward the use of this corpus for specific machine translation and computer-assisted translation applications.

The use of corpora of bilingual parallel texts seems to offer a promising tool for the future, thanks to the progress that has been made in terms of storage and computing capacities, as well as of acquisition of large amounts of text.

The idea of using parallel corpora is not new; it dates back to the early days of machine translation, but it was not used in practice until 1984 (Martin Kay 93). Subsequently, various methods have been proposed for processing the different levels of correspondence between two texts, an original and its translation.

The approach proposed here for the French-Arabic language pair (corpus-based machine translation) can be considered an extension of what was referred to, in the 1980s, as ”memory-based machine translation” (MBMT) or ”example-based machine translation” (EBMT)1. It is based on a statistical approach making use of probability calculations of equivalences between texts of the corpus.

This method is grounded on the conviction that there are no preestablished solutions to translation (theoretical procedures), but most possible solutions can be found in texts already translated by professionals. In other words, a large portion of a translator’s competence is encoded in the language equivalencies that can be found in already translated texts.Moreover, a bilingual corpus is richer in information about the language than a mono-

lingual corpus, since it provides situational equivalency information on the possibilities of the language system when in contact with a different linguistic system.

The approaches that we have seen so far, all use human-encoded linguistic knowledge to solve the translation problem. We will now look at some approaches that do not explicitly use such knowledge, but instead use a training corpus (plur. corpora) of already translated texts - a parallel corpus - to guide the translation process. A parallel corpus consists of two collections of documents: a source language collection, and a target language collection. Each document in the source language collection has an identified counterpart in the target language collection.

3.2.1 Algorithm

Corpus-based Machine Translation makes use of past translation examples to generate the translation of a given input. An EBMT system stores in its example base of translation examples between two languages, the source language and the target language. These examples are subsequently used as guidance for future translation tasks. In order to translate a new input sentence in SL, similar SL sentence is retrieved from the example base, along with its translation in TL. This example is then adapted suitably to generate a translation of the given input. It has been found that EBMT has several advantages in comparison with other MT paradigms (Sumita and Iida, 1991).

An overall idea of corpus based machine translation :

1. Split the problem into sub problem
2. Recall how they solve similar sub-problems in the past
3. Adapt these solution to the new situation
4. Combine the solution to solve the bigger problem

Corpus based Machine Translation entails three steps :

1. Matching fragments against the parallel corpus
2. Adapting the matched fragments to the target language
3. Recombining these translated fragments appropriately

3.2.2 Examples

In corpus based approach there are some samples of sentences in database:

1. Bangla: tara kheliteche

English: They are playing.

2. Bangla : krisokera dhan khete kaj koriteche

English: The farmers are working in the paddy field

3. Bangla: balokera mathe kheliteche

English: The boys are playing in the field

Now using the corpus based approach,

Source language: balokera dhan khete kheliteche

Target language: The boys are playing in the paddy field

3.2.3 Difficulties of Corpus-based Machine Translation

1. Can not use in general translation
2. But improvable by increasing Knowledge Base
3. Match sentence rule is very difficult
4. No tools available

3.3 Transfer Approach

The second variant of the indirect approach is called the transfer method. Although there is some kind of 'transfer' in any translation system, the term transfer method applies to those which have bilingual modules between intermediate representations of each of the two languages. These representations are language-dependent: the result of analysis is an abstract representation of the source text (this could be something like a phrase-structure tree). In turn, the input to generation is an abstract representation of the target text (again, possibly a tree). The function of the bilingual transfer modules is to convert source language (intermediate) representations into target language (intermediate) representations, as shown in the figure below. Since these representations link separate modules (analysis, transfer, generation), they are also frequently referred to as interface representations.

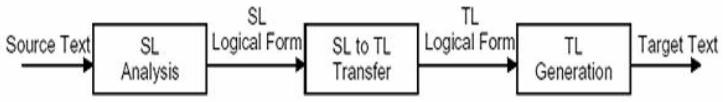


Figure 3.2: Transfer approach

Procedures:

(1) Bangla analysis (ambiguities are resolved)

(2) Bangla-English transfer (performed by a French-English bilingual module)

(3) English generation (English text generated)

In the transfer approach there are therefore no language-independent representations.In comparison with the interlingua type of multilingual system there are clear disadvantages in the transfer approach. The addition of a new language involves not only the two modules for analysis and generation, but also the addition of new transfer modules, the number of which may vary according to the number of languages in the existing system. For example, in the case of a two-language system, a third language would require four new transfer modules.

Why then is the transfer approach so often preferred to the interlingua method?

The first reason is that it is far too difficult to devise language-independent representations (interlingua).

The second is that in the transfer approach the analysis and generation grammars work between two languages and are not so difficult to write. In contrast to that in the interlingua approach these grammars are language-independent and must work for any language of the system. A little illustration will help appreciate the difference. A grammar between Ukrainian and Russian is easy to write. A grammar between Ukrainian and English is more difficult to devise. A grammar between Ukrainian and Japanese is even more difficult to formulate. Now suppose you have to write a grammar that will work for Russian AND English AND Japanese! This grammar will certainly prove to be the most difficult one to write.

Finally, if the design is optimal, the work of transfer modules can be greatly simplified and the creation of new ones can be less difficult than might be imagined.

For a multilingual MT system, a separate transfer component is required for each direction of translation for every pair of languages that the system handles. For a system that handles all combinations of n languages, n analysis components, n generation components, and n(n - 1) transfer components are required. If the transfer stage can be done away with, say by ensuring that each analysis component produces the same language-independent representation, and that each generation component produces the translation from this very representation, then n(n-1) translation systems can be provided by creating just n analysis components and n generation components.

3.3.1 Algorithm and Examples

The transfer approach involves three stages:

- Analysis
- Transfer
- Generation

Analysis stage:

The source language sentence is parsed , the sentence structure and the constituents of the sentence are identified.

Example:

Bangla: ami vat khai

Words: ami ,vat ,khai

Sentence structure: [subject] [object] [verb]

Transfer stage: Transformations are applied to the source language parse tree to convert the structure to that of the target language.

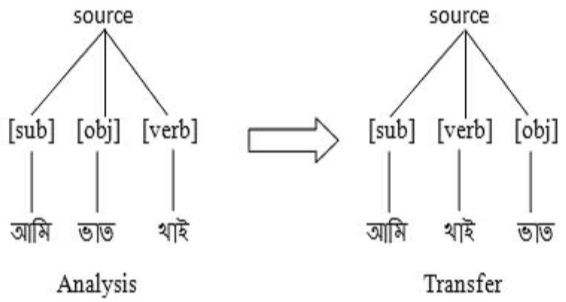


Figure 3.3: convertion from analysis to transfer stage

The Generation stage:

Translate the words and expresses the tense, number, gender etc. in the target language.

Examples:

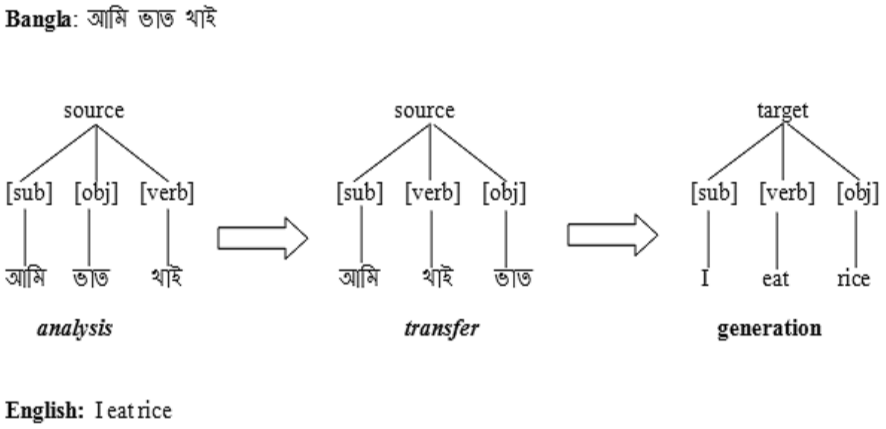


Figure 3.4: convertion from transfer stage to generation stage

3.3.2 Difficulties of Transfer approach

In comparison with the interlingua type of multilingual system there are clear disadvantages in the transfer approach. The addition of a new language involves not only the two modules for analysis and generation, but also the addition of new transfer modules, the number of which may vary according to the number of languages in the existing system. For example, in the case of a two-language system, a third language would require four new transfer modules.

3.4 Direct Approach

ONE of the earliest approaches to the Machine Translation is direct method. The Direct MT system is based upon exploitation of syntactic similarities between more or less related natural languages. Although its deficiencies soon became apparent, it remains popular in certain situations due to its usefulness, robustness and relative simplicity. One of such situation is machine translation of closely related languages. The general opinion is that it is easier to create an MT system for a pair of related languages (Hajic et.al. 2000). In the last decade, some of the systems utilizing this approach for translating between similar languages have confirmed this concept. In this paper our attempt to use the same concept for language pair of Bangla-English is described.

The direct approach lacks any kinds of intermediate stages in translation processes: the processing of the source language input text leads 'directly' to the desired target language output text. In certain circumstances the approach is still valid today - traces of the direct approach are found even in indirect systems - but the first direct MT systems had a more primitive software design.

A direct MT system is designed in all details specifically for one particular pair of languages in one direction, e.g. Bangla as the language of the original texts, the source language, and English as the language of the translated texts, the target language. Source texts are analysed no more than necessary for generating texts in the other language.

First generation direct MT systems began with what we might call a morphological analysis phase. In this phase the system identified word endings and reduced inflected forms to their uninflected basic (canonical) forms. Then it input the results into a large bilingual dictionary look-up program. There would be no analysis of syntactic structure or of semantic relationships! In other words, when the system would find the canonical form of a word, it would look it up in the bilingual dictionary to find an equivalent in the target language. There would follow some local reordering rules to give more acceptable target language output, perhaps moving some adjectives or verb particles, and then the target language text would be produced.

The direct approach is summarized in the figure below

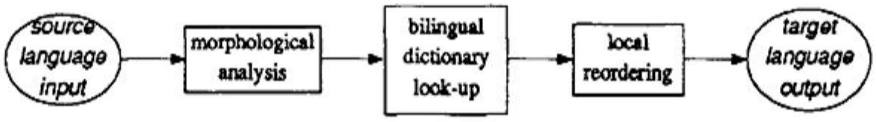


Figure 3.5: Direct Approach

Consider the sentence,

amra dese shanti chaibo

To translate this to English, we do not need to identify the thematic roles universal concepts. We just need to do morphological analysis, identify constituents, reorder them according to the constituent order in English (SVO with pre-modifiers), lookup the words in a Bangla -English dictionary, and inflect the English words appropriately! There seems to be more to do here, but these are operations that can usually be performed more simply and reliably.Direct translation systems differ from transfer and interlingua systems in that they do minimal structural and semantic (meaning) analysis.

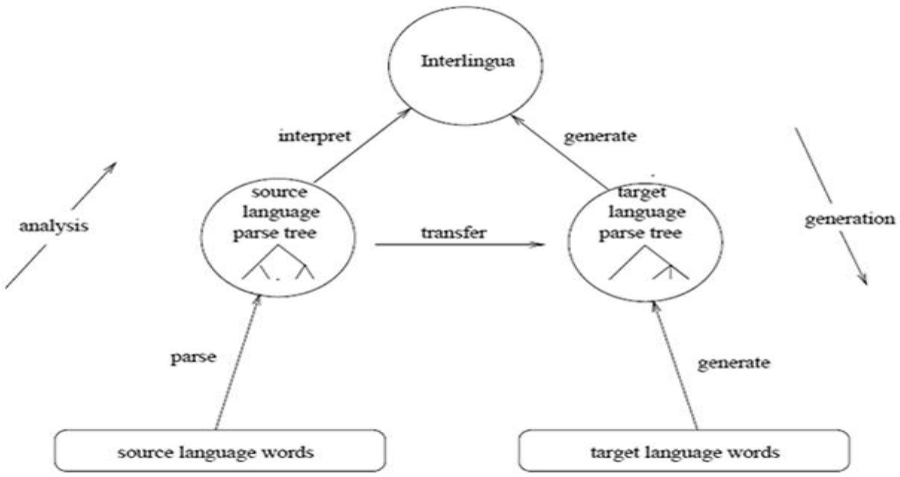


Figure 3.6: Vauquois Pyramid

Of course, the direct approach is rather ad hoc and not considered a good long term solution for MT. The linguistically more sophisticated interlingua and transfer methods are the way to go, albeit at a higher initial cost.

3.4.1 Algorithm

Step 1: Morphological analysis

Step 2: Identify constituents

Step 3: Reorder them according to the constituent order in target language.

Step 4: Lookup the words in an source-target language dictionary

Step 5: Inflect the target language words appropriately.

3.4.2 Examples

Source language: amra deshe shanti chaibo

Table 3.2 Database of Direct approach

Bangla Sentence	amra deshe shanti chaibo
Morphological Analysis	amra deshe shanti chai (vobishot kal)
Constituent Identification	(amra) (deshe) (shanti) (chai (vobishot kal))
Reorder	(amra) (chai(vobishot kal)) (shanti) (deshe)
Dictionary Lookup	We demand FUTURE peace in the country
Inflect	We will demand peace in the country

Target language : We will demand peace in the country

3.4.3 Difficulties of Direct approach

The severe limitations of this approach should be obvious. It can be characterized as 'word-for-word' translation with some local word-order adjustment. It gave the kind of translation quality that might be expected from someone with a very cheap bilingual dictionary and

only the most rudimentary knowledge of the grammar of the target language: frequent mis-translations at the lexical level and largely inappropriate syntax structures which mirrored too closely those of the source language.

3.5 Statistical Machine Translation approach

Statistical MT models take the view that every sentence in the target language is a translation of the source language sentence with some probability. The best translation, of course, is the sentence that has the highest probability. The key problems in statistical MT are: estimating the probability of a translation, and efficiently finding the sentence with the highest probability. The rest of this report provides a detailed introduction to the basic statistical MT model.

3.5.1 Statistical Machine Translation overview

One way of thinking about MT is using the noisy channel metaphor. If we want to translate a sentence f in the source language F to a sentence e1 in the target language E, the noisy channel model describes the situation in the following way:

We suppose that the sentence f to be translated was initially conceived in language E as some sentence e. During communication e was corrupted by the channel to f. Now, we assume that each sentence in E is a translation of f with some probability, and the sentence that we choose as the translation (e) is the one that has the highest probability. In mathematical terms [Brown et al, 1990],

$$e = \arg\max_{e} P(e/f) \qquad (3.1)$$

Intuitively, P(e—f) should depend on two factors:

1. The kind of sentences that are likely in the language E. This is known as the language model -P(e).

2. The way sentences in E get converted to sentences in F. This is called the translation model - P(f/e).

This intuition is borne out by applying Bayes' rule in equation 3.1:

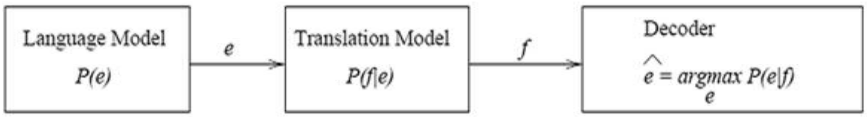


Figure 3.7: The Noisy Channel Model for Machine Translation

$$e = \arg\max_{e} \frac{p(e)p(f/e)}{p(f)} \tag{3.2}$$

Since f is fixed, we omit it from the maximization and get 3.3.

$$e = \arg\max_{e} p(e)P(f/e) \tag{3.3}$$

This model for MT is illustrated in Figure 3.4

Why not calculate P(e/f) directly as in 3.1, rather than break P(e/f) into two terms, P(e) and P(f/e), using Bayes' rule. The answer has to do with the way our translation model works, that is, the way we calculate P(e/f) or P(f/e). Practical translation models work by giving high probabilities to P(f/e) or P(e/f) when the words in f are generally translations of the words in e. Also, they do not usually care about whether the words in e go together well. For example, given the sentence,

bristy hoiteche

hoiteche bristy

to be translated to English, the translation model might give equal probabilities to the following sentences,

it is raining

This problem is circumvented if we use equation 3.3 instead of 3.1. This is because, though P(f—e) would be the same for the three sentences, the language model would rule out the last two, that is, the first translation would receive a much higher value of P(e) than the other two.

This leads to another perspective on the statistical MT model: the best translation is the sentence that is both faithful to the original sentence and fluent in the target language [Jurafsky and Martin,2000]. This is shown in equation 3.4. What we mean in terms of the above example is that both ”bristy hoiteche” and ”hoiteche bristy” may be considered equally faithful to ”It is raining” by some translation model, but the former must prove more fluent according to the language model, and should be chosen as the correct translation.

$$e = \arg\max_{e} p(e)\, p(f/e) \tag{3.4}$$

Now we have a way of finding the best translation given a set of candidate translations using 3.4, but what are these candidate translations? Unlike in our earlier supposition, we cannot practically consider each sentence in the target language. Therefore, we need a heuristic search method that can efficiently find the sentence with (close to) the highest probability. Thus, statistical translation requires three key components:

1. Language model
2. Translation model
3. Search algorithm

We take up first two components in turn in the next couple of chapters. The last problem is the standard decoding problem in AI, and variants of the Viterbi and A algorithms are used in statistical MT to solve this problem.

3.5.2 Language Modeling using N-grams

Language modeling is the task of assigning a probability to each unit of text. In the context of statistical MT, as described in the previous chapter, a unit of text is a sentence. That is, given a sentence e, our task is to compute P(e).

For a sentence containing the word sequence w1w2 . .wn, we can write without loss of generality,

P(e) = P(w1w2 . . .wn) = P(w1)P(w2/w1)P(w3/w1w2) . . . P(wn/w1w2 . . .wn-1) (3.5)

The problem here, and in fact in all language models, is that of data sparsity. Specifically, how do we calculate probabilities such as P(wn/w1w2 . . .wn-1)? In no corpus will we find instances of all possible sequences of n words; actually we will find only a miniscule fraction of such sequences. A word can occur in just too many contexts (history of words) for us to count off these numbers. Thus, we need to approximate the probabilities using what we can find more reliably from a corpus. N-gram models provide one way of doing this.

The Bi-gram approximation

In an N-gram model [Jurafsky and Martin,2000], the probability of a word given all previous words is approximated by the probability of the word given the previous N-1 words. The approximation thus works by putting all contexts that agree in the last N-1 words into one equivalence class. With N = 2, we have what is called the bigram model.

Though linguistically simple-minded, N-grams have been used successfully in speech recognition, spell checking, part-of-speech tagging and other tasks where language modeling is required. This is because of the ease with which such models can be built and used. More sophisticated models are possible-for example, one that gives each sentence structure a certain probability (using a probabilistic grammar). Such models, however, are much more difficult to build, basically because of the non-availability of large enough annotated corpora. N-gram models, on the other hand, can work with raw corpora, and are easier to build and use. N-gram probabilities can be computed in a straightforward manner from a corpus. For example, bigram probabilities can be calculated as in equation 3.6.

$$p(w_n/w_{n-1}) = \frac{count(w_{n-1}w_n)}{\sum_w count(w_{n-1}w)} \quad (3.6)$$

Here count $w_{n-1}w_n$ denotes the number of occurrences of the the sequence $w_{n-1}w_n$. The denominator on the right hand side sums over all word w in the corpus - the number of times w_{n-1} occurs before any word. Since this is just the count of w_{n-1}, we can write 3.6 as,

$$p(w_n/w_{n-1}) = \frac{count(w_{n-1}w_n)}{count(w_{n-1}w)} \tag{3.7}$$

For example, to calculate the probability of the sentence, ”all men are equal”, we split it up as,

$$\text{P(all men are equal)} = \text{P(all/start) P(men/all) P(are/men) P(equal/are)} \tag{3.8}$$

where start denotes the start of the sentence, and P(all/start) is the probability that a sentence starts with the word all.

Given the bigram probabilities in table 3.3, the probability of the sentence is calculated as in 3.9.

$$\text{P(all men are equal)} = 0.16 \times 0.04 \times 0.20 \times 0.08 = 0.00028 \tag{3.9}$$

Now consider assigning a probability to the sentence, ”all animals are equal”, assuming that the sequence ”all animals” never occurs in the corpus. That is, P(animals/all) = 0. This means that the probability of the entire sentence is zero! Intuitively, ”all animals are equal” is not such an improbable sentence, and we wouldlike our model to give it a non-zero probability, which our bigram model fails to do.

Table 3.3 Bigram probabilities from a corpus

Bigram	probability
START all	0.16
all men	0.09
men are	0.24
are equal	0.08

This brings us back to the problem of data sparsity that we mentioned at the beginning of this chapter. We simply cannot hope to find reliable probabilities for all possible bigrams from any given corpus, which is after all finite. The task of the language model, then, is not just to give probabilities to those sequences that are present in the corpus, but also to make reasonable estimates when faced with a new sequence.

Mathematically, our model in equation 3.7 uses the technique called Maximum Likelihood Estimation. This is so called because given a training corpus T, the model M that is gener-

ated is such that P(T/M) is maximized. This means that the entire probability mass is distributed among sequences that occur in the training corpus, and nothing remains for unseen sequences. The task of distributing some of the probability to unseen or zero-probability sequences is called smoothing.

Smoothing

The simplest smoothing algorithm is add-one. The idea here is to simply add one to the counts of all possible sequences - those that occur and also those that do not occur - and compute the probabilities accordingly. In terms of our bigram model, this means that the probability in equation 3.7 is now,

$$p^*(w_n/w_{n-1}) = \frac{count(w_{n-1}w_n)+1}{count(w_{n-1})+V} \tag{3.10}$$

where V is the number of word types in the vocabulary. Equation 3.10 means that we consider each possible bigram to have occurred at least once in the training corpus. Sequences that do not occur in the corpus will thus have a count of one, and consequently a non-zero probability.

In practice, add-one is not commonly used as it ends up giving too much of the probability mass to the sequences that do not occur. This is because the number of N-gram sequences that can occur in any language is a very small fraction of the number of sequences possible (all combinations of two words).

In our earlier example with unsmoothed bigrams, our problem was that the sentence, ”All animals are equal” got zero probability because the sequence ”all animals” never occurred in the training corpus. This problem is solved by add-one smoothing because now ”all animals” has a count of one and consequently some non-zero probability. Note, however, that actually impossible sequences such as ”the are” and ”a animals” and other possible but still unlikely sequences are also given a count of one. Thus add-one smoothing does not result in a realistic language model.

3.5.3 Translation Modeling

The role of the translation model is to find P(f/e), the probability of the source sentence f given the translated sentence e. Note that it is P(f/e) that is computed by the translation model and not P(e/f). The training corpus for the translation model is a sentence-aligned parallel corpus of the languages F and E. It is obvious that we cannot compute P(f/e) from counts of the sentences f and e in the parallel corpus. Again, the problem is that of data sparsity. The solution that is immediately apparent is to find (or approximate) the sentence translation probability using the translation probabilities of the words in the sentences. The word translation probabilities in turn can be found from the parallel corpus. There is, however, a glitch - the parallel corpus gives us only the sentence alignments; it does not tell us how the words in the sentences are aligned. A word alignment between sentences tells us exactly how each word in sentence f is translated in e. shows an example alignment1. This alignment can also be written as (1, (2, 3, 6), 4, 5), to indicate the positions in the English sentence with which each word in the English sentence is aligned. How to get the word alignment probabilities given a training corpus that is only sentence aligned? This problem is solved by using the Expectation-Maximization (EM) algorithm.

Expectation Maximization: The Intuition

The key intuition behind EM is this: If we know the number of times a word aligns with another in the corpus, we can calculate the word translation probabilities easily. Conversely, if we know the word translation probabilities, it should be possible to find the probability of various alignments.

Apparently we are faced with a chicken-and-egg problem! However, if we start with some uniform word translation probabilities and calculate alignment probabilities, and then use these alignment probabilities to get (hopefully) better translation probabilities, and keep on doing this, we should converge on some good values. This iterative procedure, which is called the Expectation-Maximization algorithm, works because words that are actually translations of each other, co-occur in the sentence-aligned corpus.

In the next section, we will formalize the above intuition. The particular translation model that we will look at is known as IBM Model 1 [Brown et al., 1993].

IBM Model 1

Before going on to the specifics of IBM model 1, it would be useful to understand translation modeling in a general way. The probability of a sentence f being the translation of the sentence e can be written as,

$$\text{p(f/e)} = \sum_a p(f, a/e) \tag{3.11}$$

The right hand side in equation 3.11 sums over each way (alignment) in which f can be a translation of e.The goal of the translation model is to maximize P(f—e) over the entire training corpus. In other words, it adjusts the word translation probabilities such that the translation pairs in the training corpus receive high probabilities.

To calculate the word translation probabilities, we need to know how many times a word is aligned with another word. We would expect to count off these numbers from each sentence pair in the corpus. But, each sentence pair can be aligned in many ways, and each such alignment has some probability. So, the word-alignment counts that we get will be fractional, and we have to sum these fractional counts over each possible alignment. This requires us to find the probability of a particular alignment given a translation pair. This is given by,

$$\text{p(a/f,e)} = \frac{p(f,a/e)}{p(f/e)} \tag{3.12}$$

Substituting from equation 3.11 into 3.12, we have,

$$\text{p(a/f,e)} = \frac{p(f,a/e)}{\sum_a p(f,a/e)} \tag{3.13}$$

Since we have expressed both P(a/f, e) and P(f/e) in terms of P(f, a/e), we can get a relation between the word translation probabilities and the alignment probabilities by writing P(f, a/e) in terms of the word translation probabilities and then maximizing P(f/e). Translation models essentially differ in the way they write P(f, a/e).One general way of writing P(f, a/e) is,

$$\text{p(f,a/e)=p(m/e)}\Pi_{j=1}^{m} p(a_j/a_{1,j-1}, w_{1,j-1}^{f}, m, e)\, p(w_j^f/a_{1,j}, w_{1,j-1}^{f}, m, e) \tag{3.14}$$

This equation is general except that one word in f is allowed to align with at most one position in e. Words in f can also be aligned with a special null position in e indicating that these words have no equivalent in sentence e. An example of such words is case-markers in

Hindi, which sometimes have no equivalent in English. Equation 3.17 says that given the sentence e, we can build the sentence f in the following way:

1. Choose the length m of f

2. For each of the m word positions in f

(a) Choose the position in e for this position in f. This depends on the positions already chosen, the words already chosen, m, and e.

(b) Choose the word in f in this position. This depends on the positions already chosen (including the position for this word), the words already chosen, m, and e.

IBM Model 1 is derived from this by making the following simplifying assumptions:

1. P(m—e) is a constant independent of e and m

2. A word in f has the same probability of being aligned with any position, That is,

$$p(a_j/a_{1,j-1}, w^f_{1,j-1}, m, e)=\frac{1}{l+1}$$

3. The choice of a word depends only on the word with which it is aligned, and is independent of the words chosen so far, m, and e. That is,

$$p(w^f_j/a_{1,j}, w^f_{1,j-1}, m, e)=t(w^f_j/w^e_{aj})$$

where $t(w^f_j/w^e_{aj})$ is the translation probability of w^f_j given w^e_{aj} the translation probability of the word in f in the jth position given the word in e with which it is aligned in alignment a. Now, given a parallel corpus of aligned sentences, we proceed in the following way to estimate the translation probabilities.

1. Start with some values for the translation probabilites, $t(w_f/w_e)$.

2. Compute the (fractional) counts for word translations

3. Use these counts in to re-estimate the translation probabilities.

4. Repeat the previous two steps till convergence.

This iterative use is the EM algorithm, as mentioned earlier. [Brown et al., 1993] shows that any initialization (without zero probabilities) of the $t(w_f/w_e)$ parameters leads the above algorithm to converge to the maximum.

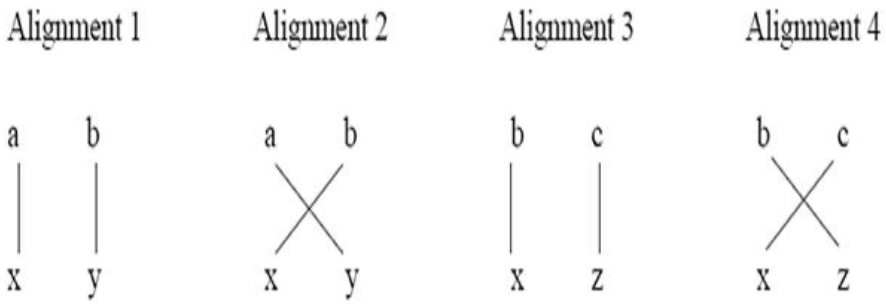


Figure 3.8: Possible word alignments in the parallel corpus

Examples of IBM Model 1

To estimate translation values from two sentences :

Example:

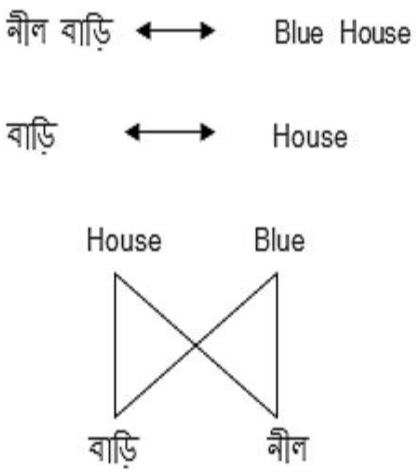


Figure 3.9: Examples of IBM Model 1

From the first example,

বাড়ি ⟷ Blue

বাড়ি ⟷ House

নীল ⟷ Blue

নীল ⟷ House

From the another,

বাড়ি ⟷ House

Step 1: Set the parameter values uniformly

t (নীল |House)= ½

t(বাড়ি|House)= ½

t(নীল|Blue)= ½

t(বাড়ি|Blue)= ½

Figure 3.10: Examples of IBM Model 1

Iteration I:

Step 2: Compute P(a,f|e) for all alignments

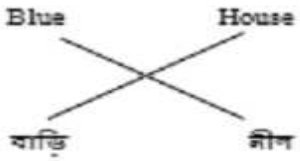
Alignment 1:

P(a,f|e)= t(বাড়ি|Blue) * t(নীল|House)

= ½ * ½

= ¼

Alignment 2:



P(a,f|e)= t(নীল|Blue) * t(বাড়ি|House)

= ½ * ½

= ¼

Alignment 3:

Figure 3.11: Examples of IBM Model 1

P(a,f|e)= t(বাড়ি|House)

$= \frac{1}{2}$

Step 3: Normalize P(a,f|e) values to yield p(a|e,f) values

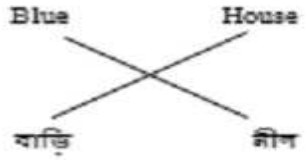
Alignment 1:

$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$

$= \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{4}}$

$= \frac{1}{2}$

Alignment 2:



$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$

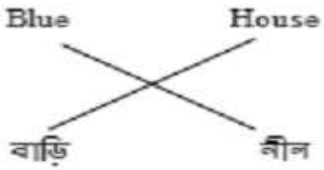
Figure 3.12: Examples of IBM Model 1

P(a,f|e)= t(বাড়ি|Blue) * t(নীল| House)

= ½ * ¼

$=\frac{1}{8}$

Alignment 2:



P(a,f|e)= t(নীল| Blue) * t(বাড়ি| House)

= ½ * ¾

$=\frac{3}{8}$

Alignment 3:

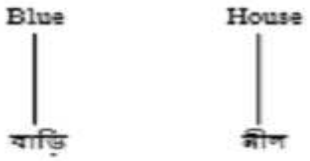
P(a,f|e)= t(বাড়ি| House)

= ¾

Figure 3.13: Examples of IBM Model 1

Step 3: Normalize P(a,f|e) values to yield p(a|e,f) values

Alignment 1:

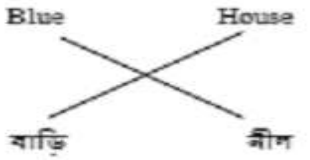


$$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

$$= \frac{\frac{1}{8}}{\frac{1}{8}+\frac{1}{8}}$$

$$= 1/4$$

Alignment 2:



$$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

$$= \frac{\frac{1}{8}}{\frac{1}{8}+\frac{1}{8}}$$

$$= 3/4$$

Figure 3.14: Examples of IBM Model 1

Alignment 3:

$$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

$$= \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{4}}$$

$$= 1$$

Step 4: Collect fractional counts

t_e (নীল |House)= ¼

t_e (নীল|Blue)= ¼

t_e (বাড়ি|Blue)= ¼

t_e (বাড়ি|House)= ¼ + 1 $= \frac{5}{4}$

Step 5: Normalize Fractional counts to get revised parameter values

$$t(\text{বাড়ি}|\text{Blue}) = \frac{\frac{1}{4}}{\frac{1}{4}+\frac{1}{4}}$$

$$= ¼$$

Figure 3.15: Examples of IBM Model 1

t (নীল |House)= $\frac{\frac{1}{4}}{\frac{1}{4}+\frac{2}{4}}$

= $\frac{1}{3}$

t(নীল|Blue)= $\frac{\frac{3}{4}}{\frac{1}{4}+\frac{3}{4}}$

= ¾

t(বাড়ি|House)= $\frac{\frac{2}{4}}{\frac{1}{4}+\frac{2}{4}}$

= $\frac{2}{3}$

Repeating the steps 2-5 times yields,

t (নীল |House) = 0.0001

t(বাড়ি|House) = 0.9999

t(নীল|Blue) = 0.9999

t(বাড়ি|Blue) = 0.0001

So, we can calculate translation probabilities by the way of these alignment probabilities.

Figure 3.16: Examples of IBM Model 1

3.5.4 Difficulties

Statistical machine translation is a very wide area but this approach is too much complex and it needs a large number of probability calculation. There are three approaches of translation model which creates a confusion about which approach to pick.

3.6 The Interlingua Approach

The interlingua method where the source text is analysed in a representation from which the target text is directly generated. The intermediate representation includes all information necessary for the generation of the target text without 'looking back' to the original text. This is an abstract representation of the target text as well as a representation of the source text. It is neutral between two or more languages. In the past, the intention or hope was to develop a representation which was truly 'universal' and could thus be intermediary between any natural languages. At present, interlingual systems are less ambitious. The interlingua approach is clearly most attractive for multilingual systems. Each analysis module can be independent, both of all other analysis modules and of all generation modules (see the figure below). Target languages have no effect on any processes of analysis; the aim of analysis

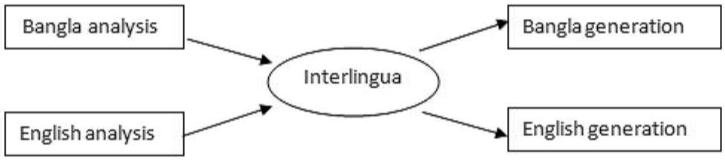


Figure 3.17: Interlingua approach

is the derivation of an 'interlingual' representation. The advantage is that to add a new language to the system one needs to create just two new modules: an analysis grammar and a generation grammar. It expresses the complete meaning of any sentence by using a set of universal concepts and relations.

3.6.1 Algorithm

The interlingua approach considers MT as a two stage process:

1. Extracting the meaning of a source language sentence in a language-independent form
2. Generating a target language sentence from the meaning.

3.6.2 Examples

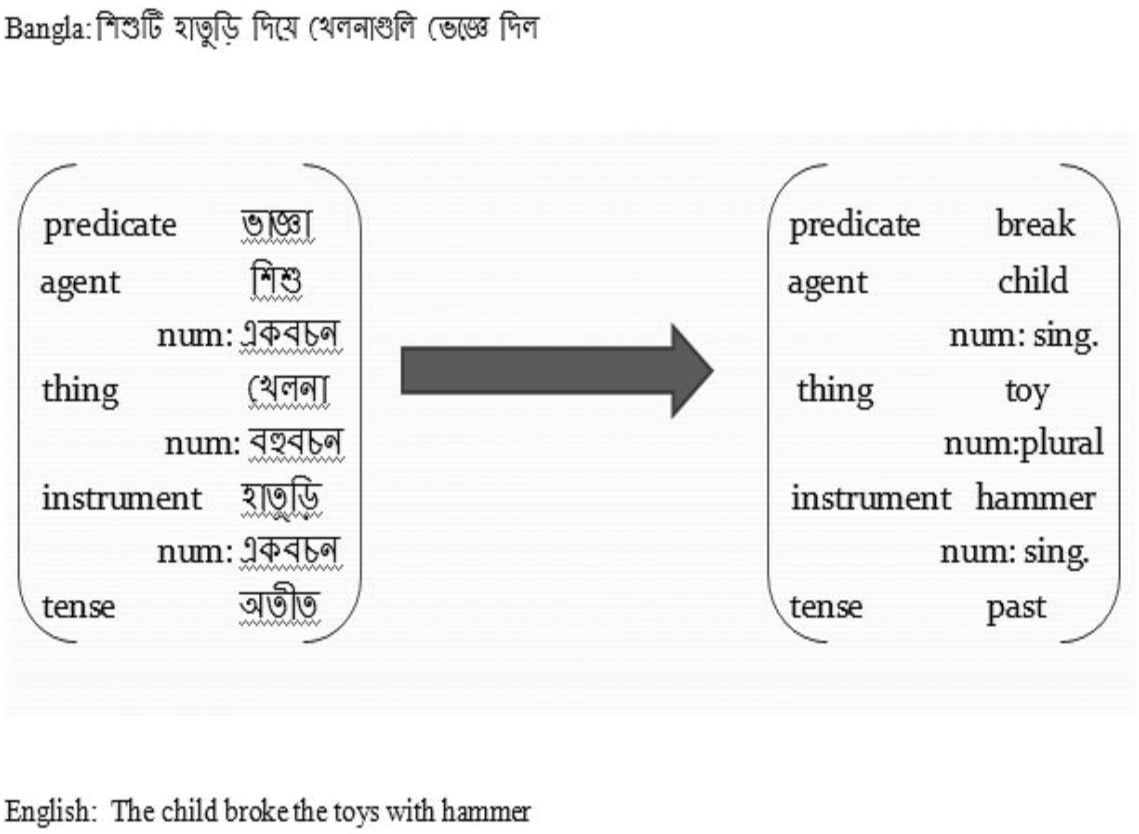


Figure 3.18: Examples of Interlingua approach

3.6.3 Difficulties of Interlingua approach

There are major disadvantages to the interlingual approach. The main is the difficulty of creating an interlingua, even for closely related languages (e.g. the Romance languages: French, Italian, Spanish, Portuguese). A truly ’universal’ and language-independent interlingua hasn’t been created so far.

CHAPTER 4
NEW IDEAS

Owing to the fact that linguistic transformation helps an MT to produce better quality target language translation.MT researcher started to develop methods to capture and process the linguistics of sentences. This was when the era of second and current generation MT systems started. Second generation MT systems are called indirect systems. In such systems the source language(SL) translation is then generated form of the text. So far we have discussed before there are six basic approaches of MT. Among all of them most widely used current MT approach is Statistical Machine Translation(SMT).

SMT models takes the view that every sentence in the target language is a translation of SL sentence with some probability. The best translation of sentence is that which has the highest probability. if t-target language and s-source language then we can write,

$$P(t/s)=p(s/t)P(t)/P(s)$$

p(t/s) depends on the P(t) which is probability of the kind of sentences that are likely to be in the language T. This known as the language model P(t).

The way sentences in s get converted to the sentences t is called translation model p(t/s).

SMT requires three major components:

- Language Model
- Translation Model
- Search Algorithn

Search algorithm for the SMT is the standard decoding problem in AI and variants of the Viterbi and A* algorithm. For rigorous implementation of this one would have to perform an exhaustive search by going all strings in the native language. Performing all the search

efficiently is the work of a MT decoder that uses the foreign string, heuristics and other methods to limit the search space and the same time keeping acceptable quality.

Implementation of translation model and language model both needs a wide variety of computation and both are complex unit to implement and extent. So we decided to work on basic other approaches rather than improving and implementing SMT as Bangla to English MT.

First comes word for word approach. At first we split the sentences into source language words and then uses a bilingual dictionary to get the outputs of the corresponding words in TL and merge this words.

The problem is that the output is not grammatically correct. Another problem is that for every word here needs every time search in the bilingual dictionary that is very time consuming and increasing the code complexity.

Next comes the direct approach.For direct approach we need a morphological analysis for identifying the tense of the verb. Then split the sentence to identify constituent. Then reorder them according to the TL. With the help of the dictionary we get the corresponding translation of the words. Using the inflection we get the tense of the verb. Whose limitation is very low quality translation and there is very frequent mistranslation at the lexical level and largely inappropriate syntax structures which mirror too closely to those the SL language.

Next comes Interlingua approach. Here SL are represented as interlingua representation where sentences are divided in predicate, agent ,theme ,instrument and tense. Finding the corresponding translation of this divides into part then we rearrange them according to the structure. A problem is that a truly universal and language independent interlingua has not been created so far. Creation of this Interlingua is very difficult even the languages are very closely related.

Next comes corpus based machine translation (CBMT) approach. Here two parallel corpora are available in SL and TL where sentences is aligned. First it is done by matching fragments against the parallel corpus and then adopting the method to the target language. Finally reassembling these translated fragments appropriately and then translation principle are applied. Here CFG is used to fix the alignment. It has been found that CBMT has several advantages in comparison with other MT paradigms.(sumita and idea 1991)

- It can be upgrading easily by adding more examples to the corpus base.
- It utilizes the translators expertise and adds reliability factor of the translation.
- It can be accelerated easily by indexing and parallel computing.

Even other researcher have considered CBMT to be one major and effective approach different MT paradigms.(KIT et. Al.2002)

So we decided to work on CBMT as we stated earlier that CBMT uses CFG for applying translation rules but in acquiring grammatical knowledge, one can acquire general CFG rules from the same annotate corpus. This kind of freedom has not been existed in the conventional method of manual knowledge creation. previously one creates if one creates a CFG grammar, it is very hard to produce a lexicalized dependency grammar based on the CFG grammar and vice versa.

So we propose a new idea for solving the CFG problem in corpus based approach. We can use the language model of the SMT for the alignment of sentences in corpus based approach. It is better than the SMT because SMT needs translation model but corpus based does not need translation model. Using the language model in CBMT time consumption becomes low and code efficiency is increased. Even the implementation becomes easier than implementing a SMT.

The last approach is the Transfer approach which works in three stages. At first in analysis stage the SL is parsed , the sentence structure (s-o-v) and the constituents of the sentences are identified. In next stage that is transfer stage, transfer is applied to the SL parse to convert the structure to that of TL .

Figure 4.1: convertion from analysis to transfer stage

In the generation stage where the words of the SL is translated and expressed in tense, number, gender etc. in TL.

The advantage of transfer approach is that the analysis and generation grammars work between two languages and are not so difficult to write. The outputs are found by applying grammatical rules and there is no need to extra alignment.

The only difficulty here is that for translating words from SL to TL its need to search from the dictionary each time for each word. So the dictionary searching is time consuming and increases the code complexity.

That's why our new proposal is to use translation model of SMT along with the transfer approach. It decreases the time consumption and increases the code efficiency.

Different approaches have been applied to translation models but there is additional complexity due to different sentence length and word order in the language. Basic three translation model approaches are Word based, syntax based, phrase based translation. In present the syntax based is used most widely. As syntax based translation is based on the idea of translating syntactic units rather than single words or string of words. So where we need to implement parse tree of sentences which consists of a complexes units of coding and due to the grammatical defferences between bangle and English it is complex task to implement and further extend as syntax based . next comes the phrase based where the aim is reduce the restriction of word based translation by translating the whole sequences of word where the lengths may differ. The sequences of words are called blocks or phrases but typically are not linguistic phrases but phrases found using statistical methods from corpora. It has been shown that restricting the phrases to linguistic phrases decrease the quality of translation.

For our Bangla to English translation we use word based translation stage because we need each word translation.

But in Bangle to English translation each word in bangle could produce any number of English word -sometimes none at all. But there is no way to group two bangle words producing a single English word. So we can use IBM model for the betterment of translation. Now a days IBM model 5 is used but its number of computation is high and it also does not solve the problem stated above. *So here we need some modification to IBM model so that it became workable for bangle sentences too. We can use IBM model 1 because its number of*

computation is relatively low and is easy to implement.. Again in IBM (2-5) we need extra probability computation for arrangement of more than one TL word for only one SL word. In IBM model 1 we will use a extra corpus where Bangla and English both sentences will be aligned. It will increase the correctness of translation and reduce the extra probability computation.

But still this technique will increase efficiency but some sentences will remain where we may get partially wrong output in auxilary verb with respect to Person.

CHAPTER 5
EXPERIMENTAL RESULT

For the implementation of our new idea we have worked with the three initial basic approaches of machine translation. These approaches are Word for word approach, Corpus based approach, Transfer based approach.

5.1 Word for word Machine Translation

In word for word approach, we need a bilingual dictionary in the database. In our experiment, the bilingual dictionary (Bangla to English) that we have used is given below :

Ami	I
amra	we
bhat	rice
bol	ball
jao	go
khai	eat
kheli	play
kothai	where
pankori	drink
pani	water
tumi	you
valo	good
kharap	bad
chele	boy
meye	girl

pochondo	like
putul	doll
khelna	toy
bari	house
ache	have
boro	large
choto	small
kukur	dog
mansho	meat
biral	cat
amr	I

Sample of inputs and outputs:

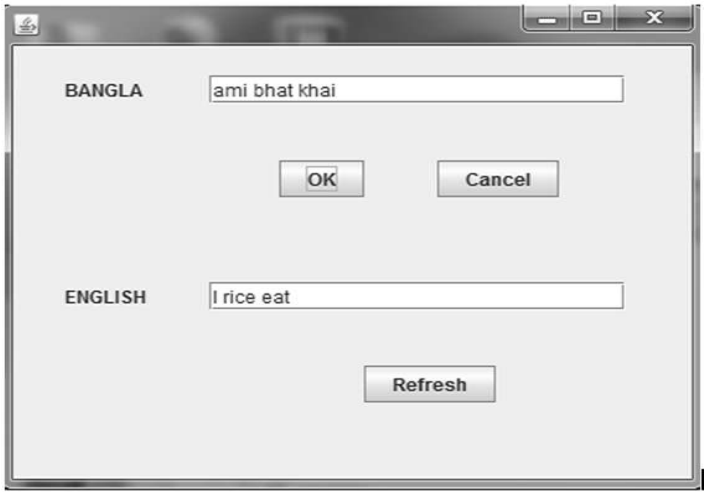


Figure 5.1: Sample of input and output of word for word approach

Figure 5.2: Sample of input and output of word for word approach

Figure 5.3: Sample of input and output of word for word approach

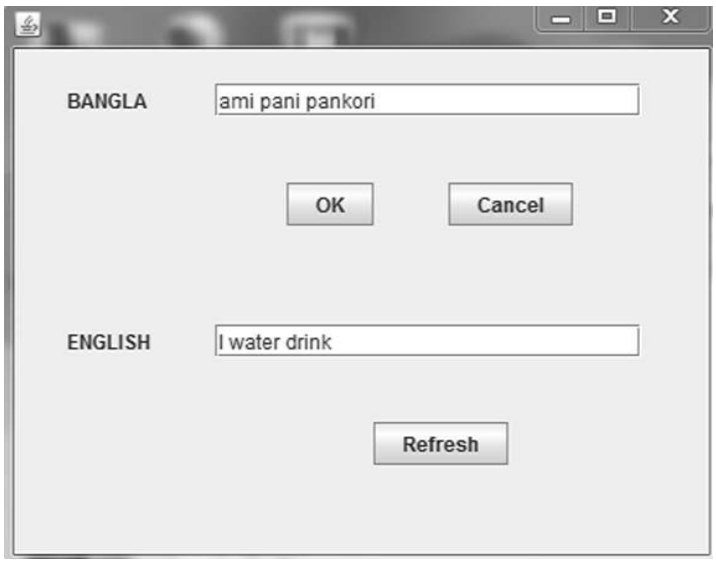


Figure 5.4: Sample of input and output of word for word approach

Figure 5.5: Sample of input and output of word for word approach

5.2 Corpus Based Machine Translation

Samples of sentences in database:

1. Bangla: Tara kriket kheliteche.

English: They are playing .

2. Bangla : krisokera dhan khete kaj koriteche

English: The farmers are working in the paddy field

3. Bangla:balokera mathe kheliteche

English: The boys are playing in the field

Samples of inputs and inputs:

Figure 5.6: Sample of input and output of corpus based approach

Figure 5.7: Sample of input and output of corpus based approach

Figure 5.8: Sample of input and output of corpus based approach

Figure 5.9: Sample of input and output of corpus based approach

Figure 5.10: Sample of input and output of corpus based approach

5.3 Transfer Based Machine Translation

1. Objects: vat, futbol, skole
2. Verb: khai, kheli, khele, jai, khao, khelo, khachi, khelche
3. Subject: ami, tumi, se, tara

Sample of inputs and inputs:

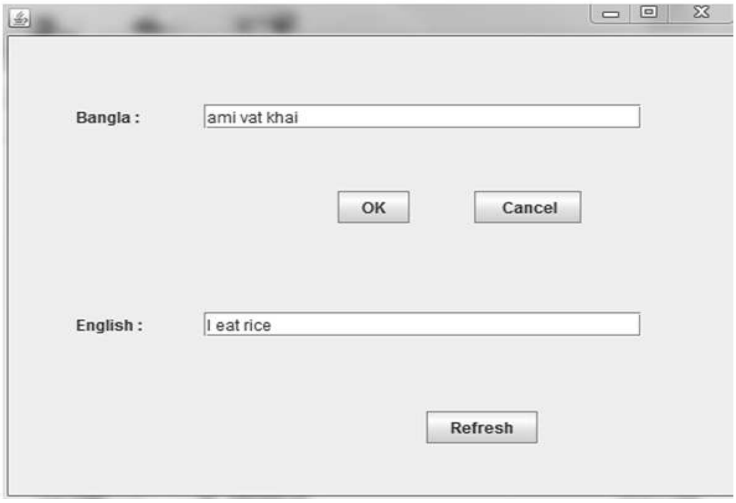


Figure 5.11: Sample of input and output of transfer based approach

Figure 5.12: Sample of input and output of tranfer based approach

Figure 5.13: Sample of input and output of tranfer based approach

Figure 5.14: Sample of input and output of tranfer based approach

Figure 5.15: Sample of input and output of tranfer based approach

CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

The field of Machine Translation is very wide and large. But Bangla to English Machine Translation System is not widely experimented and researched according to its need. We have taken an attempt to work in Bangla to English MT with the help of some basic approaches of MT. But still some improvement is necessary in our implementation. We have to include AVRO with our code for giving input in bangla sentences. Besides we have to make our implementation universal for increasing the accuracy. Although we have used IBM Model 1,which uses the aligned database for the sentences but still there are some problems here.We have to do further studies in order to take an attempt to reduce those lackings.Therefore if we have got some more time, we can improve our experimental result for increasing accuracy and efficiency.

REFERENCES

[1] Prof. Pushpak Bhattacharyya and Dr. M. Sasikumar. Statistical machine translation.

[2] Peter F. Brown. The mathematics of statistical machine translation: Parameter estimation.

[3] Magdalena Cieslak. The scope and limits of machine translation.

[4] JUDITH FRANCISCA. Adapting rule based machine translation from english to bangla.

[5] Orie Fukutomi. Experiment report of a commercial machine translation in a manufacturing industry domain.

[6] John Hutchins. Uses and applications of machine translation.

[7] W.John Hutchins. Machine translation: A brief history.

[8] Dr. Mumit Khan. Example based english to bengali machine translation.

[9] Shankar Kumar and William Byrne. Minimum bayes-risk decoding for statistical machine translation.

[10] Ph.D. Mathieu Guidre. Toward corpus-based machine translation for standard arabic.

[11] Abu Hena Mustafa Kamal Mohammad Gias Uddin, Humaid Ashraf and Muhammad Masroor Ali. New parameters for bangla to english statistical machine translation.

[12] Ananthakrishnan Ramanathan. Statistical machine translation.

[13] Harold L. SOMERS. Current research in machine translation.

[14] urpreet Singh Josan1 and Gurpreet Singh Lehal. Direct approach for machine translation.

[15] Harold L. Somers W. John Hutchins. An introduction to machine translation.

Appendix-A

Word for Word Translator

Translator.java

```
package traslator;
import javax.swing.JFrame;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Traslator {
public static void main(String[] args) {
    Translation ob= new Translation();
    ob.setVisible(true);
}}
```

Translation.java

```
package traslator;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.StringTokenizer;
public class Translation extends javax.swing.JFrame
{
public Translation()
```

```
{
initComponents();
}
private void initComponents()
{
jLabel1 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jButton3 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jLabel1.setText(”BANGLA”);
jLabel1.setName(”label1”);
jTextField1.setName(”text1”);
jButton1.setText(”OK”);
jButton1.setName(”button1”);
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton1ActionPerformed(evt);
}
}
);
```

```
jButton2.setText(”Cancel”);
jButton2.setName(”button2”);
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});
jLabel2.setText(”ENGLISH”);
jTextField2.setName(”text2”);
jButton3.setText(”Refresh”);
jButton3.setName(”button3”);
jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);
}
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGap(160, 160, 160)
```

```
.addComponent(jButton1)
.addGap(44, 44, 44)
.addComponent(jButton2))
.addGroup(layout.createSequentialGroup()
.addGap(31, 31, 31)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAUL
Short.MAX_VALUE)
.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 61, Short.MAX_VALUE))
.addGap(26, 26, 26)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(jTextField2)
.addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE, 250, Short.MAX_VALUE))))
.addContainerGap(32, Short.MAX_VALUE))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
.addContainerGap(211, Short.MAX_VALUE)
.addComponent(jButton3)
.addGap(118, 118, 118))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGap(21, 21, 21)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 18, javax.swing.GroupLayout.PR
.addGap(40, 40, 40)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton1)
.addComponent(jButton2))
.addGap(60, 60, 60)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(39, 39, 39)
.addComponent(jButton3)
.addContainerGap(54, Short.MAX_VALUE))
);
pack();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String text =jTextField1.getText();
StringTokenizer st=new StringTokenizer(text);
String string=new String();
binary ob1=new binary();
while(st.hasMoreTokens())
{
```

```
String m=st.nextToken();
String s;
ob1.search(m);
s=ob1.got;
string=string+s+” ”;
}
jTextField2.setText(string);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
jTextField2.setText(null);
}
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() )
public void run() {
new Translation().setVisible(true);
}
});
}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
```

```
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
}
```

Binary.java

```
package traslator;

import java.io.*;

import java.util.StringTokenizer;

public class binary

{

String got=new String();

public void search(String bang)

{

try

{

                FileInputStream fstream = new FileInputStream("dic.txt");

        DataInputStream in = new DataInputStream(fstream);

BufferedReader br = new BufferedReader(new InputStreamReader(in));

        String strLine;

        while ((strLine = br.readLine()) != null)

{

StringTokenizer at=new StringTokenizer(strLine);

String z=at.nextToken();

if(bang.equals(z))

{

while(at.hasMoreTokens())

got=at.nextToken();

break;
```

```
}
        }
        in.close();


}
catch (Exception e)
{
            System.err.println(”Error: ” + e.getMessage());
        }
}
}
```

Appendix-B

Corpus-Based Translator

Cba.java

```
package cba;

public class Cba

{

public static void main(String[] args)

{

NewJFrame ob= new NewJFrame();

ob.setVisible(true);

}

}
```

NewJFrame.java

```
package cba;

public class NewJFrame extends javax.swing.JFrame {

public NewJFrame() {

initComponents();

}

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jTextField2 = new javax.swing.JTextField();

jButton3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText(”BANGLA”);

jLabel2.setText(”ENGLISH”);

jButton1.setText(”OK”);

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText(”Cancel”);
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});
jTextField2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jTextField2ActionPerformed(evt);
}
});
jButton3.setText(”Refresh”);
jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);
}
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
.addContainerGap(223, Short.MAX_VALUE)
.addComponent(jButton3)
.addGap(106, 106, 106))
```

```
.addGroup(layout.createSequentialGroup()
.addGap(33, 33, 33)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 55, javax.swing.GroupLayout.PR
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 55, javax.swing.GroupLayout.PR
.addGap(18, 18, 18)
.addComponent(jTextField2, javax.swing.GroupLayout.DEFAULT_SIZE, 292, Short.MAX_VALUE)))
.addContainerGap())
.addGroup(layout.createSequentialGroup()
.addGap(130, 130, 130)
.addComponent(jButton1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 90, Short.MAX_VALUE)
.addComponent(jButton2)
.addGap(43, 43, 43))))
);
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
```

```
.addGap(61, 61, 61)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 14, javax.swing.GroupLayout.PR
.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(54, 54, 54)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jButton1)
.addComponent(jButton2))
.addGap(44, 44, 44)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 44, Short.MAX_VALUE)
.addComponent(jButton3)
.addGap(61, 61, 61))
);
pack();
}
private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String text =jTextField1.getText();
String textc=text;
```

```
System.out.println(”start”);
System.out.println(textc);
fileread t=new fileread();
t.read();
t.search(textc);
System.out.println(”end”);
jTextField2.setText(t.out);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
jTextField2.setText(null);
}
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new NewJFrame().setVisible(true);
}
});
}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
```

```
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
}
```

fileread.java

```
package cba;

import java.io.*;

public class fileread {

public String []arraym=null;

public int sizem;

NewJFrame obj=new NewJFrame();

public String[] data=new String[8];

public String strLine;

public String bang[][]={{”tara ”,”kheliteche ”},

{”krishokera ”,”dhan khete ”,”kaj koriteche ”},

{”balokera ”,”mathe ”,”kricket kheliteche ”},

{”tara ”,”dokane ”,”futbal ”,”meramot kore ”}};

public String eng[][]={{”they”,”are playing”},

{”farmers”,”in the paddy field”,”are working”},

{”boys”,”in the field”,”are playing cricket”},

{”they”,”in the shop”,”football”,”repair”}};

public int i=0,flag=0,size,index,flag1=0,j=1,k,m,i1=0;

public String out,s2,s3;

public String []array=null;

public void read() {

System.out.println(”in read”);

try{ FileInputStream fstream = new FileInputStream(”dic1.txt”); DataIn-
putStream in = new DataInputStream(fstream);
```

```
BufferedReader br = new BufferedReader(new InputStreamReader(in));          while
((strLine = br.readLine()) != null)    {          data[i1]=strLine;
System.out.println(i1);
        System.out.println (data[i1]);
i1++;
    }          in.close();
    }catch (Exception e){
        System.err.println(”Error: ” + e.getMessage());
    }
}
public void translate(String ba)
{ int c,r;
for(r=index,c=0;c<bang[r].length;c++)
{
if(bang[r][c].equals(ba))
{
if(out==null)
out=eng[r][c]+” ”;
else
out=out+eng[r][c]+” ”;
flag=1;
break;
}
}
}
```

```
public void search1(String s)
{
for(k=0;k<8;)
{
if((data[k]).equals(s))
{
flag=1;
out=out+data[k+1]+” ”;
break;
}
else
k=k+2;
}
if(flag==0)
{
for(k=0;k<=7;)
{
try
{
if(data[k].contains(s))
{ index=k/2;
translate(s);
break;
}
```

```
else k=k+2;
}
catch(Exception ex)
{
ex.fillInStackTrace();
k=k+2;
}
}
}
}
public void search(String s1)
{
System.out.println(”enter search”);
arraym=s1.split(” ”);
sizem=arraym.length;
int k1=0,f=0;
for(k1=0;k1<8;)
{
System.out.println(0);
System.out.println(s1);
System.out.println(data[k1+1]);
if(s1.equals(data[k1]))
{
out=data[k1+1];
```

```
System.out.println(”if”);
return;
}
else
{System.out.println(”else”);
k1=k1+2;}
System.out.println(k1);
}
System.out.println(2);
System.out.println(out);
while(f==0)
{
search1(s1);
if(flag==0)
{j++;
s2=null;
array=s1.split(” ”);
size=array.length;
s2=array[0]+” ”;
for(i=1;i<=size-j;i++)
s2=s2+array[i]+” ”;
// s1=s2;
search1(s2);
}
```

```
else
{
if(i==size)
{f=1;
return;
}
m=i;
flag=0;
j=0;
s3=array[m]+” ”;
m++;
for( ;m<size;m++)
s3=s3+array[m]+” ”;
s1=s3;
array=s1.split(” ”);
size=array.length;
search1(s3);
}
}
}
}
```

Appendix-C

Transfer Approach Translator

Direct.java

```
package direct;

public class Direct {

public static void main(String[] args) {

NewJFrame ob= new NewJFrame();

ob.setVisible(true);

}

}
```

NewJFrame.java

```
package direct;

public class NewJFrame extends javax.swing.JFrame {

public NewJFrame() {

initComponents();

}

@SuppressWarnings(”unchecked”)

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText(”Bangla :”);

jLabel2.setText(”English :”);

jButton1.setText(”OK”);

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});
```

```
jButton2.setText(” Cancel”);
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});
jButton3.setText(”Refresh”);
jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);
}
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
.addContainerGap(234, Short.MAX_VALUE)
.addComponent(jButton1)
.addGap(46, 46, 46)
.addComponent(jButton2)
.addGap(108, 108, 108))
.addGroup(layout.createSequentialGroup()
.addGap(48, 48, 48)
```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 51, javax.swing.GroupLayout.PR

.addGap(40, 40, 40)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField1)

.addComponent(jTextField2, javax.swing.GroupLayout.DEFAULT_SIZE, 311, Short.MAX_VALUE))

.addContainerGap(52, Short.MAX_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(292, Short.MAX_VALUE)

.addComponent(jButton3)

.addGap(139, 139, 139))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(53, 53, 53)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PR

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DE
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(48, 48, 48)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton2)

.addComponent(jButton1))

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 72, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DI
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(60, 60, 60)
.addComponent(jButton3)
.addGap(42, 42, 42))
);
pack();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String text =jTextField1.getText();
NewClass obj= new NewClass();
obj.search(text);
jTextField2.setText(obj.text_e);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(null);
jTextField2.setText(null);
}
private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new NewJFrame().setVisible(true);
}
});
}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2; }
```

NewClass.java

```
package direct;

import java.io.*;

public class NewClass {

public String obj_b[] = {”vat”,”futbol”,”skole”};

public String obj_e[] = {”rice”,”football”,”to school”};

public String verb_b[] = {”khai”,”khele”,”kheli”,”jai”,”khao”,”khelo”};

public String verb_e[]={”eat”,”play”,”play”,”go”,”eat”,”play”};

public String verb_ing_b[]={”khachi”,”khelche”,”khelchi”,”khache”};

public String verb_ing_e[]={”eating”,”playing”,”playing”,”eating”};

public String sub_ing[]={”ami”,”I am”,”tumi”,”You are”,”se”,”He is”};

public String sub_b[]={”ami”,”tumi”,”se”,”tara”};

public String sub_e[]={”I”,”You”,”He”,”They”};

public String[] array=null;

public String[] eng=null;

public String eng1,eng2,eng0;

public int flag=0;

public String text_e=null;

public void search(String s)

{

array=s.split(” ”);

int i;

for(i=0;i<6;i++)

{
```

```
try {
if(array[2].equals(verb_b[i]))
{
flag=1;
eng1=verb_e[i];
break;
}
} catch (Exception e) {
}
}
if(flag==0)
{
for(i=0;i<4;i++)
{
try {
if(array[2].equals(verb_ing_b[i]))
{
flag=2;
eng1=verb_ing_e[i];
break;
}
}
catch (Exception e) {
}
```

```
}
}
if(flag==2)
{
for(i=0;i<6;i=i+2)
{
if(array[0].equals(sub_ing[i]))
{
eng0=sub_ing[i+1];
break;
}
}
}
else
{
for(i=0;i<4;i++)
{
if(array[0].equals(sub_b[i]))
{
eng0=sub_e[i];
break;
}
}
}
```

```
for(i=0;i<3;i++)
{
try {
if(array[1].equals(obj_b[i]))
{
eng2=obj_e[i];
break;
}
}
catch (Exception e) {
}
}
try {
if(eng0.equals(”He”))
{
this.text_e=eng0+” ”+eng1+”s”+” ”+eng2;
}
else
this.text_e=eng0+” ”+eng1+” ”+eng2;
}
catch (Exception e) {
}
}
}
```