A survey of Data Driven Machine Translation

Submitted in partial fulfillment of the requirements
for the degree of

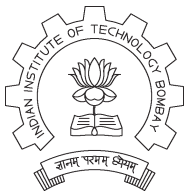
by

Somya Gupta

Roll No: 10305011

under the guidance of

Prof. Pushpak Bhattacharyya



Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Contents

List of Figures

List of Tables

Abstract

Machine Translation (MT) refers to the use of computers for translating automatically from one language to another. The differences between source and target languages and the inherent ambiguity of the source language itself make MT a very difficult problem. Traditional approaches to MT have relied on humans giving linguistic knowledge in the form of rules to transform text. Given the vastness of language, this is a highly knowledge intensive task.

Corpus-based approaches to Machine Translation (MT) dominate the MT research field today, with Example-Based MT (EBMT) and Statistical MT (SMT) representing two different frameworks within the data-driven paradigm. Example Based MT is a radically different approach that involves matching of examples from large amounts of training data followed by adaptation and recombination. This survey provides an overview of MT techniques, and covers some of the realted work in Example Based and Statistical approaches to machine translation from 1984 to 2011. The report concludes with a brief discussion on example-based hybrid techniques, existing MT systems and MT evaluation criteria.

Chapter 1

Machine Translation

Machine Translation (MT) [HS92] is defined as the process of converting a piece of text in one language to another, where the former is called the source language and the latter target language. MT is an area of research that draws ideas and techniques from linguistics, computer science, Artificial Intelligence (AI), translation theory, and statistics. Work began in this field as early as in the late 1940s, and various approaches have been tried over the past five decades.

In this chapter, we do a literature survey on Machine Translation with emphasis on the Data-Driven Machine Translation techniques. We begin by introducing the taxonomy of MT systems, followed by approaches to machine translation. Then we move on to describing Example-Based, Statistical and Hybrid techniques of machine translation in brief.

1.1 Taxonomy of MT Systems

Based on the point of entry from the source text to the target text, the taxonomy of MT systems can be illustrated by the Vauquois Triangle [Vau76] (Figure 1.1). Movements towards the top of the triangle needs deeper levels of understanding of the input texts. The three translation methodologies, viz., Direct, Transfer and Interlingua are placed at the bottom, in the middle and the top of the triangle respectively.

There are various approaches to machine translation, namely, Rule-Based or Knowledge-Driven approaches and Corpus-Based or Data-Driven approaches. The Knowledge-Driven approaches are further classified into Transfer Based and Interlingua Based MT, while the Corpus-Based approaches are classified into Example-Based and Statistical Machine Translation. The classification can be depicted as shown in Figure 1.2.

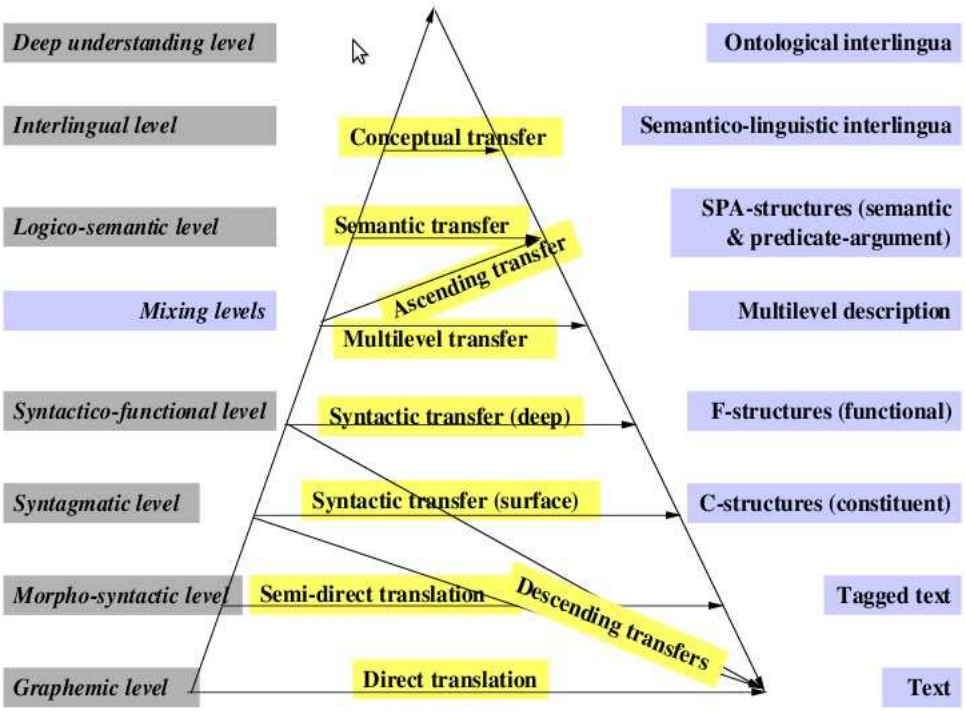


Figure 1.1: The Vauquois Triangle [Bha08]

1.2 Difficulty in MT

Although the ultimate goal of MT may be to equal the best human efforts, the current targets are much less ambitious. MT systems mostly aim to translate technical documents, reports, instruction manuals etc. The goal usually is not fluent translation, but only correct and understandable output.

The translation task is not so simple as it appears; it has many challenges like Idioms and Collocations, Polysemy, Homonymy, Synonyms Metaphors, Lexical and Structural mismatch between the languages, complicated structures, referential ambiguity and ambiguities in the source and target languages.

The Lexical and the structural mismatch are due to the difference in the way each language expresses ideas or feelings. For example, in Hindi language, the verb is inflected based on the gender of the subject in the sentence, whereas this is not so in English language. The multi-word constructs like Idioms and Collocations add more challenge in translation, as their meaning can't be derived from their constituents.

1.2.1 Lexical/Phrasal Ambiguity

Words and phrases in one language often map to multiple words in another language. For example, in the sentence,

He goes to the bank

it is not clear whether the mound of sand (तट in Hindi) sense or the financial institution (बैंक) sense is being used. This will usually be clear from the context, but this kind of disambiguation is generally non-trivial. Also, each language has its own idiomatic usages which are difficult to identify from a sentence. For example,

His grandfather kicked the bucket.

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,

The child was brought up in an orphanage. (पालना)
They brought up the table to the first floor. (ऊपर लाना)
The issue was brought up in the parliament. ((मुद्दा) उठाना)

1.2.2 Syntactic Ambiguity

Yet another kind of ambiguity is structural ambiguity, consider the following sentence:

Visiting relatives can be bad.

This can be translated in Hindi as either of the following two sentences.

रिश्तेदारों के यहां जाना बुरा हो सकता है
rishtedaaro ke yahaan jana bura ho sakta hai
visitors of place going bad be can is
or
आए हुए रिश्तेदार बुरे हो सकते हैं
aaye hue rishtedaar bure ho sakte hain
came be visitors bas be can are

depending on whether it is the relatives that are visiting are bad or going to visit the relatives is bad.

1.2.3 Semantic Ambiguity

Semantic Analysis is even more difficult to disambiguate. Consider the following two sentences:

I play with bat and ball.
I play with my friends.

These translate to the following Hindi sentences respectively:

मैं बॉल और बैट से खेलता हूँ
main ball aur bat se khelta hoon
I ball and bat with play be

मैं अपने दोस्तों के साथ खेलता हूँ
main apne dosto ke saath khelta hoon
I my friends with play be

Here, *with* in the two English sentences gets translated to से and के साथ respectively. This disambiguation requires knowledge to distinguish between bat-ball and friends.

1.3 Approaches to MT

This section discusses the various approaches to machine translation in brief.

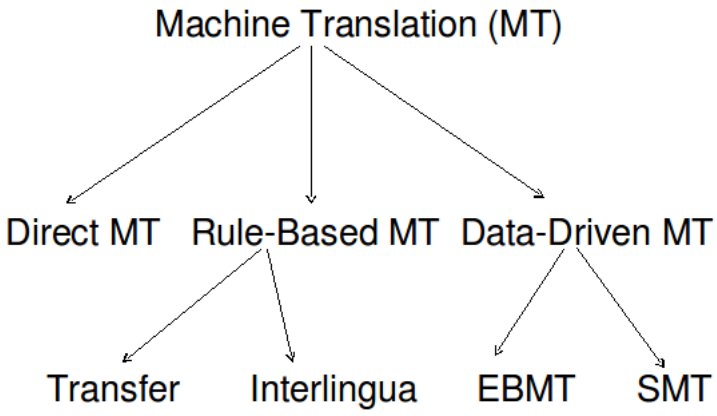


Figure 1.2: MT Approaches

1.3.1 Knowledge Driven Machine Translation

The two main knowledge-driven approaches of MT are Transfer Based MT and Interlingua Based MT. The rule-based paradigm is one of the earlest approaches to Machine Translation. It has slowly been overtaken by corpus-based techniques. The approaches within the knowledge-driven paradigm are largely driven by linguistics and rules, making use of manually created rules and resources as a basis to the translation process.

Transfer Based Machine Translation

The transfer model as shown in Figure 1.3 involves three stages: analysis, transfer, and generation. In the analysis stage, the source language sentence is parsed, and the sentence structure and the constituents of the sentence are identified. In the transfer stage, transformations are applied to the source language parse tree to convert the

structure to that of the target language. The generation stage translates the words and expresses the tense, number, gender etc. in the target language.

The stages in the translation of the sentence

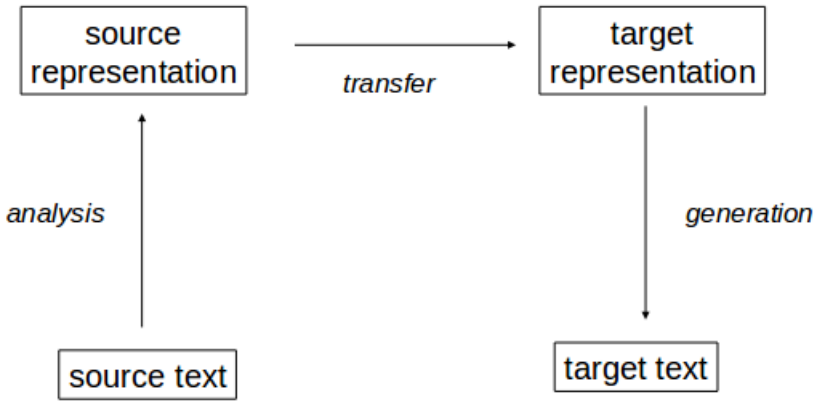


Figure 1.3: The Transfer Based Approach

There was a book on the table

The internal representation of this sentence, which is close to English structure, but with the existential there removed is

a book was on the table

The transfer stage converts this according to Hindi word order

on the table a book was

Finally, the generation stage substitutes English words with corresponding Hindi words

Hindi: मेज़ पर एक किताब थी
Transliteration: *mez par ek kitab thi*
Gloss: *table on one book was*

The analysis stage produces a source language dependent representation, and the translated output from a target language dependent representation is produced in the generation stage. Thus, 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.

Interlingua Based Machine Translation

If the transfer stage, described in the previous section, can be done away with, 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. This is precisely the idea behind the interlingua approach [DPB02].

Interlingua based models also take the source language text and constructs a parse tree. It moves one step further, and transforms the source language parse tree into a standard language-independent format, known as Interlingua. The idea of Interlingua is to represent all sentences that mean the same thing in the same way independent of language. It is to avoid explicit descriptions of the relationship between source and target language; rather it uses abstract elements, like Agent, Event, Tense, etc. The main advantage of this model is that it can be used with any language pair. The generator component for each target language takes the Interlingua as input and generates the translation in the target language.

1.3.2 Data Driven Machine Translation

The field of MT research is largely dominated by corpus-based nowadays, or data-driven approaches. The two main data-driven approaches of MT are Example- Based MT (EBMT) and Statistical Machine Translation (SMT). The corpus-based paradigm provides an alternative to direct and rule-based MT systems, The approaches within the corpus-based paradigm are largely empirical, making use of bilingual aligned corpora as a basis to the translation process.

Example Based Machine Translation

EBMT is the application of Case-Based Reasoning to MT. It is an attempt to avoid the problems of assuming a compositional transfer from source to target, translating instead by analogy. EBMT systems store a huge set of translation examples which provide coverage in context for the input.

The store of translation examples is maintained in the form of of SL TL pairs, usually aligned at the sentence level. An input sentence is matched against this repository in order to find a similar match. The identification of similarity depends on some measure of distance of meaning. The term Example-Based Translation is accredited to Nagao(1984) who introduced the notion of analogical translation [Nag84]. He stressed the following notion of detecting similarity:

The most important function is to find out the similarity of the given input sentence

and an example sentence, which can be a guide for the translation of the input sentence.

Nagao(1984) also suggested how the adaptability of a example could be checked: "the replaceability of the corresponding words is tested by tracing the thesaurus relations". If the thesaurus similarity was high enough then the example was accepted for the translation of that particular substring, if not, then another example was searched for. Example Based Machine Translation, the focus of this project, is discussed in detail in the following chapter.

Statistical Machine Translation

Statistical Machine Translation (SMT) [BCP+90] deals with automatically mapping sentences from one human language (source) to another human language (target). This process can be thought of as a stochastic process. There are many SMT variants, depending upon how translation is modeled. Some approaches are in terms of a string-to-string mapping, some use trees-to-strings, and some use tree-to-tree models. All share in common the central idea that translation is automatic, with models estimated from parallel corpora (source- target pairs) and also from monolingual corpora.

1.3.3 Hybrid Approaches

The problem with traditional approaches, i.e. the direct approach, and transfer and inter- lingua systems approaches, is that natural language expertise has to be manually encoded into their data structures and algorithms, whether as special cases or as a full representation of the conceptual content of the utterance . This has been at the expense of coverage and robustness. No technique is error-free, each has its own drawbacks. This calls for hybrid approaches to machine translation where the advantages of one technique can be clubbed with another to produce results better than an approach can produce individually.

Chapter 2

Example Based Machine Translation

EBMT is a corpus based machine translation, which requires parallel-aligned machine-readable corpora [CW03]. Here, the already translated example serves as knowledge to the system. This approach derives the information from the corpora for analysis, transfer and generation of translation. These systems take the source text and find the most analogous examples from the source examples in the corpora. The next step is to retrieve corresponding translations. And the final step is to recombine the retrieved translations into the final translation.

EBMT is best suited for sub-language phenomena like phrasal verbs, weather forecasting, technical manuals, air travel queries, appointment scheduling, *etc.* since building a generalized corpus is a difficult task. The translation work requires annotated corpus which is a very complicated task.

2.1 EBMT Architecture

The EBMT model shares similarities in structure with that shown in the Vauquois Triangle. The transfer-based model is made up of three stages: analysis, transfer and generation, as illustrated in Figure 2.1 [Som99]. In EBMT the search and matching process replaces the analysis stage, transfer is re- placed by the extraction and retrieval of examples and recombination takes the place of the generation stage. However, in Figure 2.1, 'direct translation' does not correspond exactly to 'exact match' in EBMT, as exact match is a perfect translation and does not require any adaption at all, unlike direct translation.

2.1.1 Matching

The first task in an EBMT system is to take the source-language string to be translated and to find the example (or set of examples) which most closely match it. This is

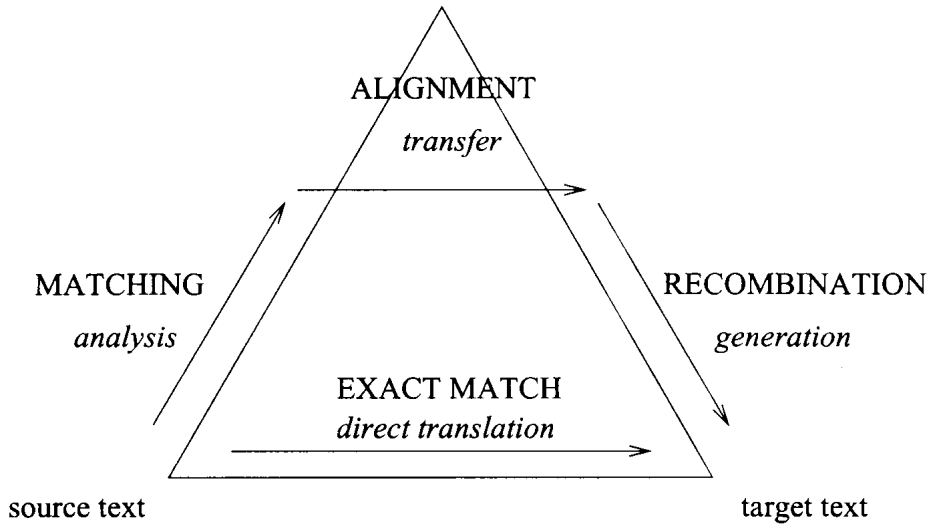


Figure 2.1: The Vauquois Triangle Modified for EBMT [Som99]

also the essential task facing a TM system. This search problem depends of course on the way the examples are stored. In the case of the statistical approach, the problems the essentially mathematical one of maximizing a huge number of statistical probabilities. In more conventional EBMT systems the matching process may be more or less linguistically motivated. Several Matching techniques are discussed in the section on Matching in EBMT.

In the matching and retrieving phase, the input text is parsed into segments of certain granularity. Each segment of the input text is matched with the segments from the source section of the corpora at the same level of granularity. The matching process may be syntactic or semantic level or both, depending upon the domain. On syntactic level, matching can be done by the structural matching of the phrase or the sentence. In semantic matching, the semantic distance is found out between the phrases and the words. The corresponding translated segments of the target language are retrieved from the second section of the corpora.

In establishing a mechanism for the best match retrieval, the crucial tasks are:

1. Determining whether the search is for matches at sentence or sub-sentence level, that is determining the "text unit"

2. The definition of the metric of similarity between two text units.

2.1.2 Alignment

Having matched and retrieved a set of examples, with associated translations, the next step is to extract from the translations appropriate fragments ("alignment" or "adaptation").

2.1.3 Recombination

Recombination means to combine the aligned fragments so as to produce a grammatical target and output ("recombination"). This is arguably the most difficult step in the EBMT Its difficulty can be gauged by imagining a source-language process: monolingual trying to use a Translation Memory system to compose a target text. The problem is twofold:

1. Identifying which portion of the associated translation corresponds to the matched portions of the source text

2. Recombining these portions in an appropriate manner

2.2 Issues in EBMT

Harold Somers, in his review article on Example Based Machine Translation [Som99] discusses some issues that need to be considered while using the example based approach to machine translation.

Parallel Corpora

Since EBMT is corpus-based MT, the first thing that is needed is a parallel aligned Machine- readable corpus. EBMT systems are often felt to be best suited to a sub language approach,and an existing corpus of translations often serve to define implicitly the sub language which the system can handle. Once a suitable corpus has been located, there remains the problem of aligning which segments (typically sentences) correspond to each other. The alignment problem can of course be circumvented by building the example database manually, as is sometimes done for TMs, when sentences and their translations are added to the memory as they are typed in by the translator.

Granularity of examples

The task of locating appropriate matches as the first step in EBMT involves a trade-off between length and similarity. As put by Nirenburg et al. (1993), The longer the matched passages, the lower the probability of a complete match. The shorter the passages, the greater the probability of ambiguity and the greater the possibility of resulting translation being of low quality. Although the sentence as a unit appears to be an obvious grain-size which is easy to determine, the matching and recombination process needs to be able to extract smaller chunks from the examples.

How many examples

The way examples are stored and used may also affect the number of examples needed by the translation system. Although it has been proved that the quality of translations improves as more examples are added to the database (Mima et al. 1998), it is assumed

that there is some limit after which further examples do not improve the quality of translations.

Suitability of examples

The assumption that an aligned parallel corpus can serve as an example database is not universally made. Several EBMT systems work from a manually constructed database of examples. There are several reasons for this. A large corpus of naturally occurring text will contain over- lapping examples of two sorts: some examples will mutually reinforce each other, either by being identical, or by exemplifying the same translation phenomenon. But other examples will be in conflict: the same or similar phrase in one language may have two different translations no other reason than inconsistency.

How the examples are stored

EBMT systems differ quite widely in how the translation examples themselves are actually stored. Obviously,the storage issue is closely related to the problem of searching for matches,discussed in one of the subsequent sections.

Annotated Tree Structures

Early attempts at EBMT - where the technique was often integrated into a more conventional rule-based system- stored the examples as fully annotated tree structures with explicit links. Planas and Furuse(1999) represent examples as a multi-level lattice, combining lexical, syntactic and other information. orthographic, Although their typographic, proposal is aimed at TMs, the approach is also suitable for EBMT.

Generalized Examples

In some systems, similar examples are combined and stored as a single "generalized" example. Brown (1999) [BCP+90] for instance tokenize the examples to show equivalence classes such as "person's name", "date","city name", and also linguistic information such as gender and number. In this approach, phrases in the examples are replaced by these tokens,thereby making the examples more general.

Statistical Approaches

In these systems, the examples are not stored at all, except in as much as they occur in the corpus on which the system is based. What is stored is the precomputed for statistical parameters which give the probabilities bilingual word pairings, the "translation model". The "language model" which gives the probabilities of target word strings being well-formed is also precomputed, and the translation process consists of a search for the target-language string which optimizes the product of the two sets of probabilities, given the source-language string.

2.3 Translation Memory

Translation memory (TM) is defined by the Expert Advisory Group on Language Engineering Standards (EAGLES) Evaluation Working Group's document on the evaluation of NLP systems as:

> a multilingual text archive containing (segmented, aligned, parsed and classified) multilingual texts, allowing storage and retrieval of aligned multilingual text segments against various search conditions.

In other words, translation memory consists of a database that stores source and target language pairs of text segments that can be retrieved for use with present texts and texts to be translated in the future.

The use of TM involves two phases[SN90]:

1. The first consists in accumulating translation units (TU). A TU is simply a source sentence (source TU) and its corresponding translation, also called the target sentence (target TU), as the human translator has translated it.

2. In the second phase, an input source sentence (INPUT) being given, the TM retrieves the more similar source TU and proposes the corresponding target TU as a close translation of the input sentence INPUT.

2.3.1 Formalizing Translation Memories

An important issue that arises while using Translation Memory in MT is what does the more similar sentence mean? Is it the number of different characters between the Input and source TU? In this case, in the example below, is sentence (0) closest to sentence (1) that has five different letters, or to sentence (2) that has seven different letters?

(0) The wily child drowned himself	शैतान बच्चे ने अपने आपको डुबा दिया
(1) The wise chief crowned himself	समझ्दार अध्यक्श ने अपने आपको ताज पहनाया
(2) The wily children drowned him-self	शैतान बच्चों ने उसे डुबा दिया

The TELA structure: Separating the data into different layers

Rather then a flat heterogeneous structure, a multilevel structure, homogeneous by level is proposed [PF99]. The levels are called "layers" and the whole structure TELA. TELA is a French acronym for "Treillis Etags et Lis pour le traitement Automatique", meaning "web" in Spanish, and standing for "Floored and Linked Lattices for Automatic Processing".

The two improvement directions considered in the paper are:

- The separation of the document data into a "layered" structure
- The inclusion of linguistic data in supplementary layers

This structure can have as many layers as necessary. Each layer is a lattice whose bottom is inferior to all elements of the layer, and top superior to all these elements'. Eight basic layers are proposed:

1. Text Characters: This layer contains all relevant characters involved in the real text.
2. Words: This is simply the sequence of the surface forms of the words of the sentence.
3. Lemmas (Basic forms): The lemmas, are part the result of shallow parsing for a precise process of the sentences.
4. Parts of Speech (POS): This is also comes from the shallow parsing.
5. XML content tags: These tags represent where to apply layout attributes in the original XML segment.
6. XML empty tags: These tags cope with objects inserted in the flow of text of the XML segment.
7. Glossary entries
8. Linguistic analysis structures: This level depends on how far the available linguistic analyzer can go.

The above matching is in fact based on an **Edit Distance**. A level f and two sentences (1) and (2) being given, we consider the layers of TELA structures T1 and T2 as sequences of items:

$$s_1^f = (s_1^f i)_{1<i<n_1^f}$$
$$s_2^f = (s_2^f i)_{1<i<n_2^f}$$

where $n_1^f <= n_2^f$, here n_1^f and n_2^f represent the number of items of the layer f of T1 and T2. The edit distance be- tween layers s_1^f and s_2^f is the total cost of the sequence of elementary operations transforming s_1^f into s_2^f that minimizes this total cost. Here are the classical elementary operations:
Equality (cost 0)
Deletion (cost 1)
Insertion (cost 1)

Similarity between two TELA structures

For matching TELA structures as in Figure 2.2, the following edition operations between the layers of T1 and T2 are considered:
Layer 1 equality (score 1)
...
Layer F equality (score 1)
Deletion for all layers (score 1)
Insertion for all layers (score 1)

$$sum^1 2_1 \ ..., \ sum^1 2_F; \ sum^1 2_- \ sum^1 2_+$$

Let the above equation be the sequence of the number of these elementary operations for editing S_2 into S_1. For example the edition of T4 into T3 has the following sequence; **2, 4, 5, 2, 1: 4, 1**

The similarity between S_1 and S_2 is defined as the following vector:

$$\sigma^{12}=(sum^{12}_1/n^1_1,...,sum^{12}_F/n^1_1, 1\text{-}sum^{12}_-/n^2_2, 1\text{-}sum^{12}_+/n^2_1)$$

	1	2	3	4	5	6
2	~~click~~	a	color	~~and~~	~~press~~	~~enter~~
3	click	a	color	~~and~~	press	~~enter~~
4	verb	art	noun	~~conj~~	verb	O
5			em		idx	
7						~~ENTER~~

Figure: Simplified TELA Structure for Sentence 1 [PF99]

	1	2	3	4	5	6	7	8	9
2	~~he~~	~~clicks~~	~~on~~	a	color	;	~~then~~	~~presses~~	~~OK~~
3	~~he~~	click	~~on~~	a	color	;	~~then~~	press	~~OK~~
4	~~pp~~	verb	~~prep~~	art	noun	;	~~conj~~	verb	O
5					em			idx	
7									~~OK~~

Figure: Simplified TELA Structure for Sentence 2 [PF99]

2.3.2 Linking Translation Memory and EBMT

EBMT and TMSs have in common the use of a database of previous translations, the 'memory' or 'example-base' and given a piece of text to translate, finding in the example database the best matches for that text. Once the match has been found, the two

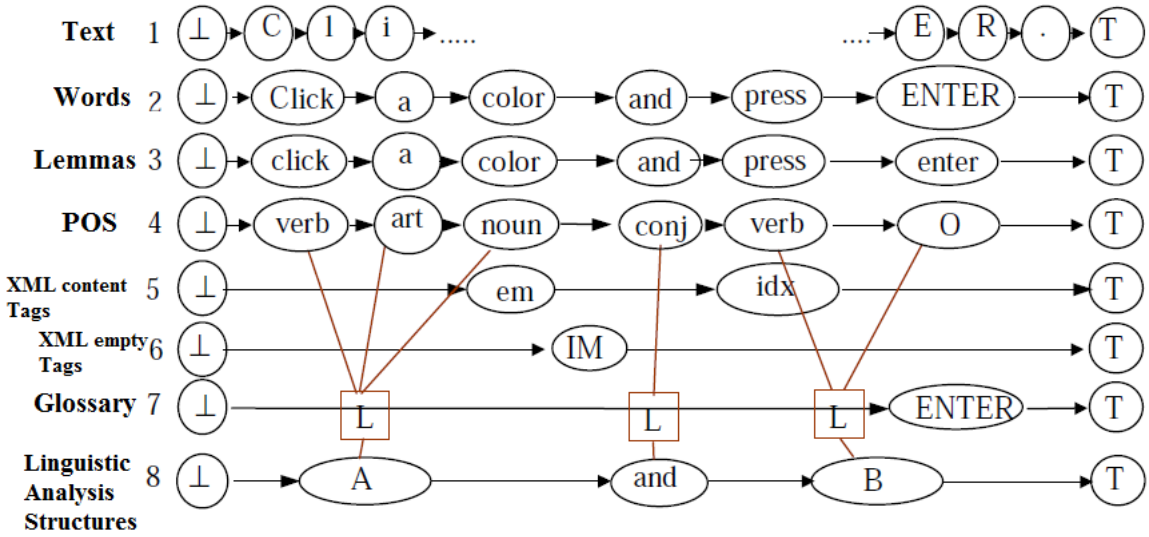


Figure 2.2: The TELA Structure [PF99]

techniques begin to diverge. [SD04]

How examples are found and stored, the matching techniques used and number of matches to be considered are common to both TMs and EBMT. However there are important differences, mainly stemming from the fact that a TMS is a translators aid, where the user has the main responsibility for making decisions, whereas EBMT is a way of doing translation automatically. The main difference then lies in the fact that a TMS has essentially just the single step of matching examples, while EBMT must then do something with the matches found. What EBMT does consists of two steps, often referred to as alignment and recombination.

2.4 Matching Techniques

2.4.1 Introduction

In the matching and retrieving phase, the input text is parsed into segments of certain granularity. Each segment of the input text is matched with the segments from the source section of the corpora at the same level of granularity. The matching process may be syntactic or semantic level or both, depending upon the domain. On syntactic level, matching can be done by the structural matching of the phrase or the sentence. In semantic matching, the semantic distance is found out between the phrases and the words. The corresponding translated segments of the target language are retrieved from the second section of the corpora.

In establishing a mechanism for the best match retrieval, the crucial tasks are:
(i) determining whether the search is for matches at sentence or sub-sentence level,

that is determining the "text unit", and
(ii) the definition of the metric of similarity between two text units.

2.4.2 EBMT using DP-matching between word sequences

The proposed approach retrieves the most similar example by carrying out DP-matching of the input sentence and example sentences while measuring the semantic distance of the words [Sum01]. Then the approach adjusts the gap between the input and the most similar example by using a bilingual dictionary. The resources used in EBMT DP Matching are shown in Figure 2.3 The translation process consists of four steps:

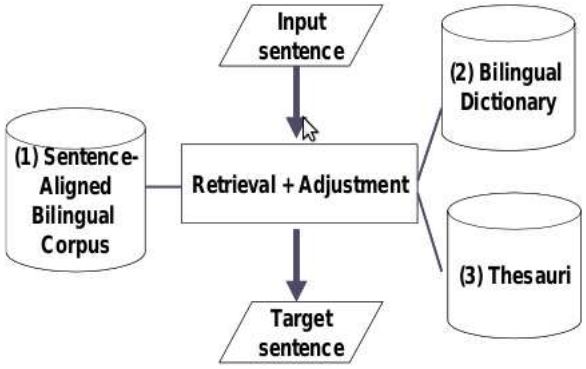


Figure 2.3: Resources used in EBMT DP Matching [Sum01]

I. Retrieve the most similar translation pair;
II. Generate translation patterns;
III. Select the best translation pattern;
IV. Substitute target words for source words.

Retrieval

This step scans the source parts of all example sentences in the bilingual corpus. By measuring the distance between the word sequences of the input and example sentences, it retrieves the examples with the minimum distance, provided the distance is smaller than the given threshold.

$$(1)\ \mathbf{dist} = \frac{I+D+2\sum SEMDIST}{L_input+L_example}$$

$$(2)\ \mathbf{SEMDIST} = \frac{K}{N}$$

In equation (1), the **dist** is calculated as follows: The counts of the **Insertion (I), Deletion (D), and Substitution (S)** operations are summed up and the total is normalized by the sum of the length of the source and example sequences.

Substitution (S) considers the semantic distance between two substituted words and is called SEMDIST. SEMDIST is defined as the division of K (the level of the least common abstraction in the thesaurus of two words) by N (the height of the thesaurus) according to equation (2) (Sumita and Iida, 1991). It ranges from 0 to 1.

Pattern Generation

First, the step stores the hatched parts of the input sentence in memory for the following translation. Second, the step aligns the hatched parts of source sentence to corresponding target sentence of the translation example by using lexical resources.

Pattern Selection

The following heuristic rule for pattern selection is used:

1. Maximize the frequency of the translation pattern.
2. If this cannot be determined, maximize the sum of the frequency of words in the generated translation patterns.
3. If this cannot be determined, select one randomly as a last resort.

Word Substitution

By translating the source word of the variable using the bilingual dictionary, and instantiating the variable within the target part of the selected translation pattern by target word, the target sentence is obtained.

2.4.3 A Matching Technique In Example-Based MT

Cranias, Papageorgiou, and Piperidis (1994) describea matching technique in example-based machine translation. To encode a sentence into a vector, information about the **functional words (fws)** appearing in it is used, as well as about the lemmas and pos tags of the words appearing between fws.[CPP94]

To identify the fws in a given corpus the following criteria are applied :

- fws introduce a syntactically standard behaviour
- most of the fws belong to closed classes.
- the semantic behaviour of fws is determined through their context
- most of the fws determine phrase boundaries
- fws have a relatively high frequency in the corpus

Algorithm

1. fws can serve the retrieval procedure with respect to the following two levels of contribution towards the **similarity score** of two sentences :
   - Identity of fws of retrieved example and input **(I)**
   - fws of retrieved example and input not identical but belonging to the same group **(G)**
   - A negative score called the penalty score **(P)**
2. Each non-fw by its **Ambiguity class (ac)** and the corresponding lemma(s) (for example, the unambiguous word "see" would be represented by the ac which is the set verb and the lemma "eat")
   - Overlapping of the sets of possible lemmas of the two words **(L)**
   - Overlapping of the ambiguity classes of the two words **(T)**
3. Whenever an I or G transition is investigated, the system calls the **second level DP-algorithm** which produces a local additional score due to the potential similarity of lemmas and tags of the words lying between the corresponding fws

The algorithm also determines, through a backtracking procedure, the relevant parts of the two vectors that contributed to the similarity score thus obtained.

2.4.4 Two approaches to matching in EBMT

Description Here.[NDG93] [3] Sergei Nirenburg, Constantine Domashnev and Dean J. Grannes:Two approaches to matching in example-based machine translation.TMI-93: The Fifth International Conference on Theoretical and Methodological Issues in Machine Translation,Kyoto, Japan

2.4.5 Other Matching Techniques

Apart from the trivial character based and word based matching techniques, and those discussed above, several other techniques of matching exist such as Carroll's Angle of Similarity, Annotated Word-based Matching, Structure-based Matching, Partial Matching for Coverage. These have been described in the review article on example-based machine translation by Somers [Som99].

2.5 Adaptation and Recombination

In the final phase of translation, the retrieved target segments are adapted and recombined to obtain the translation. It identifies the discrepancy between the retrieved

target segments with the input sentence's tense, voice, gender, etc. The divergence is removed from the retrieved segments by adapting the segments according to the input sentence's features.

In the recombination phase, direct matches are sought between source sentence chunks and the chunks in translation templates. It is possible that there are no translation templates that directly match source side text chunks bound with unmatched parts covered by the multi-chunks. In such cases, recombination cannot proceed to produce full translations as multi-chunk alignments on the target side of the template will remain uninstantiated.

A solution to this is to search for more translation templates that cover the uninstantiated parts using recursive matching. This method attempts to match an source text chunks compositionally against multiple chunks from the set of translation templates. A recursive algorithm is invoked that attempts to match successively shorter portions of the source side, against the chunks in the translation templates. The target equivalents are concatenated naively, according to the order of the matches with the portions of the source chunks.

2.6 Approaches to EBMT

Approaches to EBMT can be broadly classified into proportional analogy based, runtime approaches and template-driven example-based machine translation. These have been described in the following sections.

2.6.1 EBMT Using Proportional Analogies

A review of EBMT using proportional analogies

EBMT using proportional analogies [SDN09] is the pure st EBMT technique where statements are of the relationship between four entities as in

$$A : B :: C : D$$

In this way, treating sentences as strings of characters, proportional analogies can be handeled.

They swam in the sea : *They swam fast* :: *It floated in the sea* : *It floated fast*

For the purpose of EBMT a database of example pairs is assumed where each sentence has a corresponding translation.
Some difficulties faced in this approach:

- The first is that for a given input sentence, D, the database may contain multiple triples (A, B, C) that offer a solvable analogy

- because of the unconstrained nature of proportional analogy as a mechanism, there is always the possibility of "false analogies", that is, sets of strings for which the analogy holds, but which do not represent a linguistic relationship
- The proportional analogy method can consider the examples to be either strings of characters, or strings of words. The latter approach of course eliminates the possibility of outputs but also means that correspondences would not be captured such as

$$walks : walked :: floats : floated$$

While this approach seems fraught with difficulties as a stand-alone translation model, its use for the special case of unknown words, particularly named entities or specialist terms, seems much more promising .

Mitigating Problems in Analogy-based EBMT with SMT and vice versa

In this paper [DSMN10] the authors have implemented the EBMT system using PAs based on Lepage (1998, 2005c). They distinguish between three main components in their system. These three components are used to solve both source- and target-side analogies. The three main components of the analogy-based EBMT, namely Heuristics, Analogy Verifier and Analogy Solver, are depicted in figure 2.4:

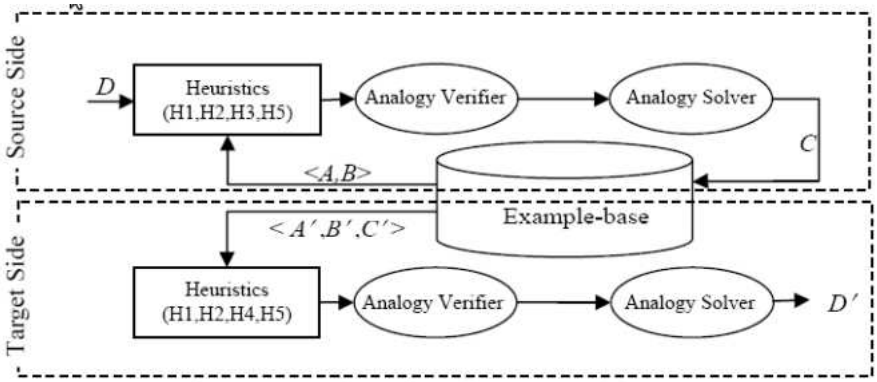


Figure 2.4: Architecture EBMT using Analogies [DSMN10]

Firstly, the system requires some knowledge about choosing relevant $< A, B >$ pairs from the example-base to ensure that the better candidate analogical equations from the potential set of all possible analogies are solved first, and also to filter out some of the unsolvable analogies before verification. Different heuristics are adopted to ensure this. Secondly, there is an Analogy Verifier, which decides the solvability of an analogical equation. The third component solves the analogy based on the triplet $< A, B, D >$ and produces C. Note that D is the input sentence to be translated. This module is called the Analogy Solver. Once C is produced in the source side, the translation equivalents $< A, B, C >$ are found in the target side for the source side

$< A, B, C >$ triplet. Further, the three components in the target side are applied in the same order to obtain one candidate translation D. Collecting all D, they are ranked by frequencies as different analogical equations might produce identical solutions.

2.6.2 Template Driven EBMT

EBMT based on the extraction and recombination of translation templates can be placed between runtime approaches to EBMT and derivation tree based EBMT. Here Translation examples serve the decomposition of text to be translated and determine the transfer of lexical values into the target language. Translation templates determine the word order of the target language and the type of phrases. An induction mechanism generalizes translation templates from translation examples.

Inducing Translation Templates for Example-Based MT

This paper describes an example-based machine translation system which makes use of morphological knowledge, shallow syntactic processing, translation examples and an induction mechanism which induces translation templates from the translation examples [Car99]. Induced translation templates determine a) the mapping of the word order from the source language into the target language and b) the type of sub-sentential phrases to be generated. Morphologic knowledge allows the abstraction of surface forms of the involved languages, and together with shallow syntactic processing and the percolation of constraints into reduced nodes new input sentences to be translated can be generalized. The generalized sentence is then specified and refined in the target language where refinement rules may adjust the translated chunks according to the target language context.

This paper investigates the computational power of the generalization process and describes in more detail the possibilities and limits of the translation template induction mechanism. Translation templates are seen as generalized translation examples. The template correctness criterion (TCC) formally defines the correctness of induced translation templates and the one tree principle implies a strict segmentation strategy.

Due to the induction capacities of the system the rule system remains relatively simple if the source and target language are structural similar. The conjunction of different resources allows for the analysis and generation of context-sensitive languages. Mapping of structurally different languages remains, however, beyond the capacities of the induction mechanism.

Learning Translations Templates from Bilingual Translation Examples

In this paper, the authors present a model for learning translation templates between two languages [CG01]. The model is based on a simple pattern matcher. This model

was integrated with an example-based translation model into Generalized Exemplar-Based Machine Translation. For this, Translation Template Learning algorithms have been proposed which eliminate the need for manually encoding the translation templates.

The translation is done by initially inferring a set of translation templates. Also, learning is done incrementally. The templates learned from the previous examples help in learning new templates from new examples. In the translation process, a given source language sentence in surface form is translated into the corresponding target language sentence in surface form. This is done through the following steps:

- First, the word level representation of input sentence is derived by using a lexical analyzer
- The translation templates matching the input are then collected. These templates are those that are most similar to the sentence to be translated. For each selected template, its variables are instantiated with corresponding values in the source sentence. Then, templates matching these bound values are retrieved and recombined.
- Lastly, the surface level representation of the sentence obtained in previous step is generated.

Sub-Phrasal Matching and Structural Templates in Example-Based MT

This work describes a system that synthesizes two different approaches to EBMT [Phi07]. A system named Cunei is explained that borrows heavily from ideas and techniques present in EBMT and PB-SMT. This system maintains the indexing scheme and sub-phrasal matching found in Panlite and adds to this a "light" version of the structural matching found in the Gaijin system.

Instead of using constituent phrases identified by the marker hypothesis, as the structure of each sentence, Cunei uses only the sequence of part-of-speech tags. This system does not require one template, it finds examples corresponding to any sub-section of the input sentence. It passes the resulting lattice to the same language modeler used by Panlite for decoding.

The limitations of the system are with combining scores from two different probability distributions, which is a hard problem. Moreover, the phrases inserted in the lattice do not always have optimal boundaries. These result in partial translations that sometimes inappropriately guide the language modeler.

2.6.3 EBMT Using Chunk Alignments

Chunk parsing was first proposed by Abney (1991). Over the years, many researchers have experimented with chunk-based alignments. Work has been done to produce Chinese chunked sentences via chunk projection from dictionary and English chunked sentence. Zhou et al. (2004) extracted chunk pairs automatically to use in an SMT system. After aligning chunks using their co-occurrence similarity, they extract chunk-pairs and report a significant improvement in translation quality.

Chunk Based EBMT

Corpus driven machine translation approaches such as Phrase-Based Statistical Machine Translation and Example-Based Machine Translation have been successful by using word alignment to find translation fragments for matched source parts in a bilingual training corpus. However, they still cannot properly deal with systematic translation for insertion or deletion words between two distant languages [KBC10]. In this work, the authors used syntactic chunks as translation units to alleviate this problem, improve alignments and show improvement in BLEU for Korean to English and Chinese to English translation tasks. In the algorithm, chunk translation sequence pairs are extracted as shown in 2.5:

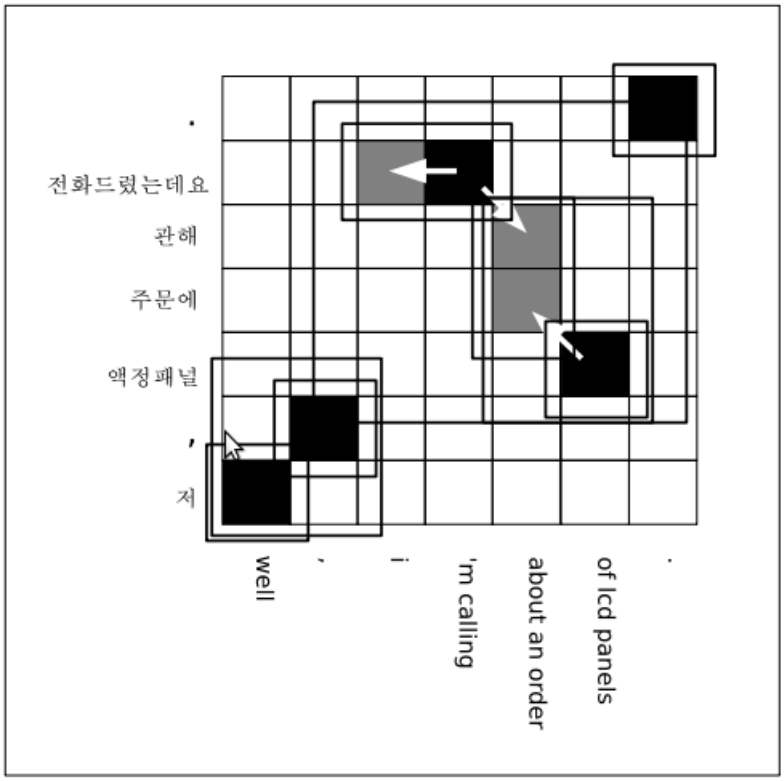


Figure 2.5: Chunk Translation Sequence pair extraction [KBC10]

2.7 Comparison to other MT Techniques

Here, we compare Example-Based Machine Translation with the other two most commonly followed approaches to MT, namely, Statistical Machine Translation (SMT) and Rule-Based Machine Translation (RBMT). We infer from the comparison that EBMT essentially takes a stance between SMT and RBMT. It tries to combine the advantages of both and avoid their limitations. The generalized template extraction is essentially an automatic rule-generation strategy where translation templates are learned from examples, as discussed previously. Moreover, being data-driven no manual task is required for generation of translations. Hence, it is a data-driven MT technique like SMT.

2.7.1 EBMT and SMT

Both SMT and EBMT are data driven methods of machine translation. Although EBMT came much earlier than SMT, the latter is the state-of-the-art approach to machine translation. Most work on machine translation in the recent years has focused mainly on SMT. Here, we list the similarities and differences between the two corpus-based approaches to machine translation.

Statistical MT	**Example Based MT**
Correlation based	Analogy/Matching based
It is an efficient method when a large corpus is available and no linguistic knowledge is available. Good when both the languages are poor morphological languages.	Used when the examples in the bilingual corpus are very much similar to the other examples present in the corpus and the input sentence is derivable from the examples present in the corpus.
The efficiency of translation depends on the quality of bilingual corpus available. Its hard to correctly estimate the probability of rare words.	The translation becomes very difficult if there is no corresponding example in the corpus. The corpus should have overlapping sentences, so as to extract translated phrase for a matched source language phrase.
A bilingual dictionary is not required	A bilingual dictionary is required
Even if the input sentence is present in the corpus there is no guaranty to have the same translation back as output	Exact translation of the input sentence is obtained.

Table 2.1: Comparison between EBMT and SMT

2.7.2 EBMT and RBMT

Here we list the similarities and differences between EBMT and Rule-Based MT. Sumita et al. 1990 give a detailed comparison of EBMT and RBMT in [SIK90]. The more elaborate RBMT becomes, the less expandable it is. Considerably complex rules concerning semantics, context, and the real world, are required in Rule-based machine translation. This is overcome by Example-Based MT. The following table compares the two MT techniques:

Basis	**Rule Based MT**	**Example Based MT**
Example Independence	No; specific to a particular system	Yes; knowledge is completely independent of the system
Measurement of reliability factor	No; RBMT has no device to compute the reliability of the result.	Yes; a reliability factor is assigned to the translation result according to the distance between input and retrieved similar example.
Robustness	High. RBMT works on exact match reasoning.	Low. EBMT works on best match reasoning.

Table 2.2: Comparing EBMT and RBMT

Chapter 3

Statisical Machine Translation

SMT models take the view that every sentence in the target language is a translation of the source language sentence with some probability. This probability is a function of both faithfulness and fluency. The best translation, of course, is the sentence that has the highest probability. If we chose the product of faithfulness and fluency as our quality metric, we could model the translation from a source language sentence S to a target language sentence T as:

$$Best\text{-}translation(T) = \underset{T}{\text{argmax}}\ faithfulness(T, S) * fluency(T)$$

This intuitive equation clearly resembles the Bayesian noisy channel model. Lets make the analogy perfect and formalize the noisy channel model for statistical machine translation. First of all, for the rest of this report, well assume we are translating from a Foreign(English) language sentence $F = f_1, f_2, ..., f_m$ to Hindi(E). In a probabilistic model, the best Hindi sentence $e = e_1, e_2, ..., e_l$ is the one whose probability $P(E|F)$ is the highest. As is usual in the noisy channel model, we can rewrite this via Bayes rule:

$$\hat{E} = \underset{E}{\text{argmax}}\ P(E|F) \tag{3.1}$$

$$= \underset{E}{\text{argmax}}\ \frac{P(F|E)P(E)}{P(F)} \tag{3.2}$$

We can ignore the denominator $P(F)$ inside the argmax since we are choosing the best Hindi sentence for a fixed Foreign sentence F, and hence $P(F)$ is a constant. The resulting noisy channel equation shows that we need two components:

- Translation Model, $P(F|E)$*[faithfulness]*
- Language Model, $P(E)$*[fluency]*

Notice that applying the noisy channel model to machine translation requires that we think of things backwards, as shown in Fig 3.1. We pretend that the Hindi input E while passing through a noisy channel is corrupted to some Foreign sentence F. Now our task is to discover the hidden sentence E that generated our observation sentence F.

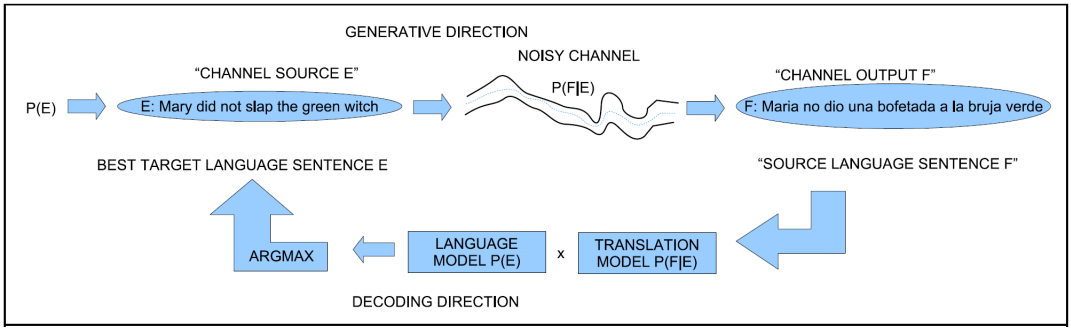


Figure 3.1: Noisy channel model for translation

The noisy channel model of statistical MT thus requires three components to translate from a French sentence F to an English sentence E:

- A language model to compute $P(E)$
- A translation model to compute $P(F|E)$
- A decoder, which is given F and produces the most probable E

We take up first two components in turn in the next couple of sections. 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.

Analogy Consider F as a set of medical symptoms and E as a disease. There are many diseases that could give rise to these symptoms. If we build a generative model, then we can reason about the probability of any disease E occurring, as well as the probability that symptoms F will arise from any particular disease E, i.e. $P(E)$ and $P(F|E)$. They may conflict: you may have a common disease that often gives rise to symptoms F, and you may have a very rare disease that always gives rise to symptoms F. Deciding it is difficult but biologists know roughly how diseases cause symptoms, i.e. $P(F|E)$, thus it is possible to build computer models showing how this happens. It is not so obvious how to build a single model that reasons from symptoms to diseases, i.e. $P(E|F)$. Furthermore, we may have independent sources of information about $P(E)$ in isolation, such as old hospital records. That is to say that, if we reason directly about translation using $P(E|F)$, then probability should be very good whereas using Bayes Rule, we can get theoretically good translations even if the probability numbers aren't that accurate. (K. Knight, 1999)

3.1 Language Models

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 section, 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 $w_1 w_2 ... w_n$, we can write without loss of generality,

$$P(E) = P(w_1 w_2 ... w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)...P(w_n|w_1 w_2 ... w_{n-1}) \quad (3.3)$$

The problem here, and in fact in all language models, is that of data sparsity. Specifically, how do we calculate probabilities such as $P(w_n|w_1 w_2 ... w_{n-1})$? In no corpus will we find instances of all possible sequences of n words. Actually we will find only a minuscule 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.

N-Gram Approximation

In an n-gram model [JM08], 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, and $N = 3$ gives the trigram model.

N-gram probabilities can be computed in a straightforward manner from a corpus. For example, bigram probabilities can be calculated as

$$P(w_n|w_{n-1}) = \frac{count(w_{n-1}w_n)}{\sum_w count(w_{n-1}w)} \quad (3.4)$$

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 that is the number of times w_{n-1} occurs before any word. Since this is just the count of w_{n-1}, we can write the above equation as,

$$P(w_n|w_{n-1}) = \frac{count(w_{n-1}w_n)}{count(w_{n-1})} \quad (3.5)$$

3.2 Translation Models

As discussed in chapter 2, 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. 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.

EM Algorithm

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 [BPPM93].

Word-based SMT

In Word Based SMT, the atomic unit of translations are word. Each word from the source language is translated to the target language and then aligned to the correct position. The most famous alignment models for these are developed by IBM [BPPM93].

IBM Model 1

The notation that is used in the following discussion is summarized in the following table

f	*the source language sentence*
e	*the target language sentence*
m	*length of* f
l	*length of* e
w^f	*a word in* f *(generally)*
w_i^f	*the ith word in* f
$w_{i,m}^f$	*the sequence* $w_1^f, w_2^f, ...w_m^f$
w^e	*aword in* e *(generally)*
w_i^e	*the* i^{th} *word in* e
$w_{i,l}^e$	*the sequence* $w_1^e w_2^e ... w_l^e$
a	*a particular alignment between* f *and* e
a_i	*the position in* e *with which the ith word in* f *is aligned*
$a_{i,m}$	*the sequence* $a_1 a_2 ... a_m$
Ψ	*all possible alignments between* f *and* e

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,

$$P(f|e) = \sum_{a \epsilon \Psi} P(f, a|e) \tag{3.6}$$

The right hand side in above equation 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,

$$P(a|f, e) = \frac{P(f, a|e)}{P(f|e)} \tag{3.7}$$

Substituting from equation (3.7) into (3.8), we have,

$$P(a|f, e) = \frac{P(f, a|e)}{\sum_{a \epsilon \psi} P(f, a|e)} \tag{3.8}$$

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,

$$P(f, a|e) = \underbrace{P(m|e)}_{\text{choose length}} \prod_{j=1}^{m} \overbrace{P(a_j|a_{1,j-1}, w^f_{1,j-1}, m, e)}^{\text{choose position}} \underbrace{P(w^f_j|a_{1,j}, w^f_{1,j-1}, m, e)}_{\text{choose word}} \tag{3.9}$$

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 3.10 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 (ϵ) independent of e and m

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

$$P(a_j|a_{1,j-1}, w^f_{1,j-1}, m, e) = \frac{1}{(l+1)} \tag{3.10}$$

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,

$$P(w^f_j|a_{1,j}, w^f_{1,j-1}, m, e) = t(w^f_j|w^e_{a_j}) \tag{3.11}$$

where $t(w^f_j|w^e_{a_j})$ is the translation probability of w^f_j given $w^e_{a_j}$ the translation probability of the word in f in the j^{th} position given the word in e with which it is aligned in alignment a.

Given these assumptions, we can write $P(f, a|e)$ in Model 1 as,

$$P(f, a|e) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(w^f_j|w^e_{a_j}) \tag{3.12}$$

Since each of the m words can be aligned with any of the $l+1$ positions in e, $(l+1)^m$ alignments are possible, summing over which we have,

$$P(f|e) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(w^f_j|w^e_{a_j}) \tag{3.13}$$

As mentioned earlier, our goal is to maximize $P(f|e)$. This is subject to the constraint that the translation probabilities for each word in e sum to 1. That is,

$$\sum_f t(w^f|w^e) = 1. \tag{3.14}$$

Introducing Lagrange multipliers $_{w^e}$, we convert the constrained maximization problem above to an unconstrained maximization problem, where we maximize the expression in equation 3.16 below.

$$h(t, \lambda) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(w^f_j|w^e_{a_j}) - \sum_{w^e} \lambda_{w^e} (\sum_f t(w^f|w^e) - 1) \tag{3.15}$$

The maximum occurs when the partial derivatives of h with respect to the components of t and are zero. The partial derivative with respect to just gives back equation 3.15. With respect to $t(w^f|w^e)$, the partial derivative is,

$$\frac{\partial h}{\partial t(w^f|w^e)} = \frac{\epsilon}{l+1^m}\sum_{a_1=0}^{l}...\sum_{a_m=0}^{l}\sum_{j=1}^{m}\delta(w^f,w_j^f)\delta(w^e,w_{a_j}^e)t(w^f|w^e)^{-1}\prod_{k=1}^{m}t(w_k^f|w_{a_k}^e) - \lambda_{w^e} \tag{3.16}$$

where δ is the Kronecker delta function, which is equal to one when both its arguments are the same and zero otherwise.
To see why the derivative is so, note that we have differentiated with respect to each word translation probability, $t(w^f|w^e)$. In each alignment where w^f and w^e are aligned (this is taken care of by the Kronecker delta), the derivative consists of all translation probabilities in that alignment, except $t(w^f|w^e)$. This is achieved by multiplying all the translation probabilities in the alignment ($\prod_{k=1}^{m}t(w_k^f|w_{a_k}^e)$) and dividing by the translation probability with respect to which we are differentiating $t(w^f|w^e)^{-1}$.
This partial derivative will be zero when,

$$t(w^f|w^e) = \lambda_{w^e}^{-1}\frac{\epsilon}{l+1^m}\sum_{a_1=0}^{l}...\sum_{a_m=0}^{l}\sum_{j=1}^{m}\delta(w^f,w_j^f)\delta(w^e,w_{a_j}^e)t(w^f|w^e)^{-1}\prod_{k=1}^{m}t(w_k^f|w_{a_k}^e) \tag{3.17}$$

Substituting from Equation 3.13 into Equation. 3.18

$$t(w^f|w^e) = \lambda_{w^e}^{-1}\sum_{a\epsilon\psi}P(f,a|e)\sum_{j=1}^{m}\delta(w^f,w_j^f)\delta(w^e,w_{a_j}^e) \tag{3.18}$$

We mentioned earlier that the word translation probabilities can be found from the fractional counts of the number of times the words are aligned with each other. Note that the counts are fractional because the alignments are not certain but probabilistic. These fractional counts can be written as,

$$c(w^f|w^e;f,e) = \sum_{a\epsilon\psi}P(f,a|e)\sum_{j=1}^{m}\delta(w^f,w_j^f)\delta(w^e,w_{a_j}^e) \tag{3.19}$$

This is the count of w^f given w^e for $(f|e)$. As expected this sums up probability of each alignment where the words co-occur.

From 3.8 and replacing $_{w^e}$ by $_{w^e}P(f|e)$, we can write the intuitive relation between the word translation probabilities and the fractional counts as,

$$t(w^f|w^e) = \lambda_{w^e}^{-1}c(w^f|w^e;f,e) \tag{3.20}$$

Since our parallel corpus contains many sentence pairs, $(f^{(1)}|e^{(1)}), (f^{(2)}|e^{(2)}), ..., (f^{(S)}|e^{(S)})$,

$$t(w^f|w^e) = \lambda_{w^e}^{-1}\sum_{s=1}^{S}c(w^f|w^e;f,e) \tag{3.21}$$

The term $\lambda_{w^e}^{-1}$ indicates that the translation probabilities should be normalized.
There is, however, another problem here. Equation 3.20 that calculates the fractional counts requires us to sum over $(l+1)^m$ alignments, which is not at all feasible. With an average sentence length of ten, we would have more than a billion alignments to go over every time! Thankfully, we can simplify this by noting that,

$$\sum_{a_1=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(w_j^f | w_{a_j}^e) = \prod_{j=1}^{m} \sum_{i=0}^{l} t(w_j^f | w_i^e) \tag{3.22}$$

Substituting this into 3.14, and then evaluating the partial derivative yields,

$$c(w^f | w^e; f, e) = \frac{t(w^f | w^e)}{t(w^f | w_0^e) + ... + t(w^f | w_l^e)} \sum_{j=1}^{m} \delta(w^f, w_j^f) \sum_{i=1}^{l} \delta(w^e, w_i^e) \tag{3.23}$$

The number of operations required now is proportional to $(l+m)$ and not $(l+1)^m$.
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 probabilities, $t(w^f | w^e)$.
2. Compute the (fractional) counts for word translations using 3.24.
3. Use these counts in 3.22 to re-estimate the translation probabilities
4. Repeat the previous two steps till convergence.

This iterative use of equations 3.24 and 3.22 is the EM algorithm, as mentioned earlier.

IBM Model 2

In Model 1, we take no cognizance of where words appear in either string. The first word in the French string is just as likely to be connected to a word at the end of the English string as to one at the beginning. In Model 2 we make the same assumptions as in Model 1 except that we assume that $P(a_j | a_1^{j-1}, w_{1,j-1}^f, m, e)$ depends on j, a_j and m, as well as on l. We introduce a set of alignment probabilities,

$$a(a_j | j, m, l) = P(a_j | a_1^{j-1}, w_{1,j-1}^f, m, e) \tag{3.24}$$

which satisfy the constraints

$$\sum_{i=0}^{l} a(i | j, m, l) = 1 \tag{3.25}$$

Hence the final equation becomes

$$P(f|e) = \epsilon \sum_{a_1=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(w_j^f | w_{a_j}^e) a(a_j | j, m, l) \tag{3.26}$$

IBM Model 3

IBM model 3 is much more advance than the previous models. It introduces the concept of fertility which allows the mapping of one source word to multiple target word. The generative model has 5 steps.

1. For each English word w_i^e , we choose a fertility ϕ_i. The fertility is the number of (zero or more) Foreign words that will be generated from w_i^e , and is dependent only on w_i^e .

2. We also need to generate Foreign words from the NULL English word. Instead of having a fertility for NULL, we'll generate spurious words differently. Every time we generate an English word, we consider (with some probability) generating a spurious word (from NULL).

3. We now know how many Foreign words to generate from each English word. So now for each of these Foreign potential words, generate it by translating its aligned English word. As with Model 1, the translation will be based only on the English word. Spurious Foreign words will be generated by translating the NULL word into Foreign.

4. Move all the non-spurious words into their final positions in the Foreign sentence.

5. Insert the spurious Foreign words in the remaining open positions in the Foreign sentence.

Model 3 has more parameters than Model 1. The most important are the n, t, d and $p1$ probabilities. The fertility probability of a word w_i^e is represented by the parameter n. So we will use $n(1|green)$ to represent the probability that English green will produce one Foreign word, $n(2|green)$ is the probability that English green will produce two Foreign words, $n(0|did)$ is the probability that English did will produce no Foreign words, and so on. Like IBM Model 1, Model 3 has a translation probability $t(w_j^f|w_i^e)$. Next, the probability that expresses the word position that English words end up in in the Foreign sentence is the distortion probability, which is conditioned on the English and Foreign sentence lengths. For example ,the distortion probability $d(1, 3, 6, 7)$ expresses the probability that the English word w_1^e will align to Foreign word w_3^f , given that the English sentence has length 6, and the Foreign sentence is of length 7. As we suggested above, Model 3 does not use fertility probabilities like $n(1|NULL)$, or $n(3|NULL)$ to decide how many spurious Foreign words to generate from English NULL. Instead, each time Model 3 generates a real word, it generates a spurious word for the target sentence with probability $p1$. This way, longer source sentences will naturally generate more spurious words.

In order to compute $P(f, a|e)$, well need to multiply the main three factors n, t and d for generating words, translating them into Foreign, and moving them around. So a first pass at $P(f, a|e)$ would be:

$$\prod_{i=0}^{l} n(\phi_i|w_i^e) * \prod_{j=0}^{m} t(w_j^f|w_{a_j}^e) * \prod_{j=0}^{m} d(j|a_j, l, m) \tag{3.27}$$

But this isn't sufficient as it stands. We need to add factors for generating spurious words, for inserting them into the available slots, and a factor having to do with the number of ways (permutations) a word can align with multiple words. Next equation gives the true final equation for IBM Model 3.

$$P(f, a|e) = \overbrace{\binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0}}^{\text{generate spurious}} * \underbrace{\frac{1}{\phi_0!}}_{\text{insert spurious}} * \overbrace{\prod_{i=0}^{l} \phi_i!}^{\text{multi-align permutations}}$$
$$* \prod_{i=0}^{l} n(\phi_i|w_i^e) * \prod_{j=0}^{m} t(w_j^f|w_{a_j}^e) * \prod_{j=0}^{m} d(j|a_j, l, m) \tag{3.28}$$

Once again, in order to get the total probability of the Foreign sentence we'll need to we would sum over all possible alignments:

$$P(f|e) = \sum_a P(f, a|e) \tag{3.29}$$

Phrase-based SMT

Motivation

In Phrase based alignment models [KOM03], the source sentence is segmented into phrases. Depending on the Alignment algorithm used, phrases need not strictly be linguistically motivated, they can be any contiguous sequence of words from the sentence i.e. N-grams. Each phrase is translated into target output phrases and finally the target phrases are reordered using a reordering model to generate the required output. Similar to the word alignment models, phrase alignment models are composed of phrase pairs and their associated probability values. During the training phase, translation probabilities are learned over phrase pairs and the target sentence is formed by subsequent phrase translations from the source sentence broken into a sequence of phrases during decoding. Advantages of Phrase Based Translation over Word-Based Translation are:

- Handling of many-to-many translations.

- Use of contextual information
- Dependency on surrounding words can be captured and translation probabilities are calculated using surrounding words.

Phrase Alignment Models

The probability model for phrase-based translation relies on a translation probability and a distortion probability. The factor $\phi(\bar{w_j^f}|\bar{w_i^e})$ is the translation probability of generating Foreign phrase $\bar{w_j^f}$ from English phrase $\bar{w_i^e}$. The reordering of the Foreign phrases is done by the distortion probability d. Distortion in statistical machine translation refers to a word having a different ('distorted') position in the Foreign sentence than it had in the English sentence. The distortion probability in phrase-based MT means the probability of two consecutive English phrases being separated in Foreign by a span (of Foreign words) of a particular length. More formally, the distortion is parameterized by $d(a_i - b_{i-1})$, where a_i is the start position of the Foreign phrase generated by the i^th English phrase $\bar{w_i^e}$, and b_{i-1}) is the end position of the Foreign phrase generated by the $(i-1)^{th}$ English phrase $\bar{w_{i-1}^e}$. We can use a very simple distortion probability, in which we simply raise some small constant α to the distortion.

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \tag{3.30}$$

This distortion model penalizes large distortions by giving lower and lower probability the larger the distortion.

The final translation model for phrase-based MT is:

$$P(f|e) = \prod_{i=1}^{l} \phi(\bar{w_j^f}|\bar{w_i^e}) d(a_i - b_{i-1}) \tag{3.31}$$

Factor-based SMT

Factored Translation model embeds the linguistic knowledge in phrase translation model. Phrase translation model are purely statistical and doesn't used any linguistic knowledge during phrase alignment. Adding morphological, syntactic or semantic information during pre-processing and post-processing steps can be valuable. One of the drawbacks of pure statistical approach is its inability to handle different word forms with same stem like house and houses. Occurrence of house in the training data does not add anything to the knowledge about houses. Hence if there is no occurrence of houses, it would be treated as an unknown word during translation despite the numerous instances of house. Such a problem occurs more frequent in morphologically rich languages like Hindi. Factored model can tackle this by translating lemma and morphological information separately and then combine the information to produce the correct target language word. This approach of translating on the level of lemmas is very desirable for morphologically rich languages.

Motivation

As learnt above, one of the drawbacks of pure statistical approach is its inability to handle different word forms with same stem like house and houses. Occurrence of house in the training data does not add anything to the knowledge about houses. Hence if there is no occurrence of houses, it would be treated as an unknown word during translation despite the numerous instances of house. Such a problem occurs more frequent in morphologically rich languages like Hindi. Factored model can tackle this by translating lemma and morpholog- ical information separately and then combine the information to produce the correct target language word. This approach of translating on the level of lemmas is very desirable for morphologically rich languages.

3.3 Decoding

In the decoding (translation) phase, the system (Moses) tries to translate all possible segmentations of the input English sentence using the phrase mapping that was learned in the training phrase. Moses tries various reordering of the Hindi phrases, and using beam search, find the translation which has the maximum probability (approximately) based on the phrase-mapping probabilities, the statistical reordering model, and the language model. Briefly the steps taken by Moses can be categorized as

- Beam Search
- Hypothesis Expansion
- Hypothesis Combination

Chapter 4

Hybrid Machine Translation

4.1 The MT Model Space

The MT Model Space is shown in Figure 4.1 The example-based x-axis represents

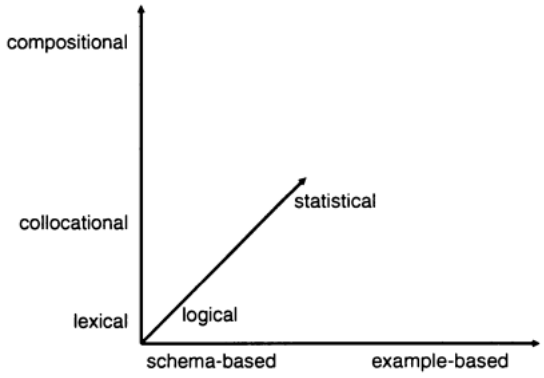


Figure 4.1: The MT Model Space

the degree to which abstraction (generalization or adaptation) is performed during testing as opposed to during training. The compositional y-axis represents the degree to which rules are compositional as opposed to lexical. The statistical z-axis represents the degree to which models make appropriate use of statistics as opposed to logic-based or set theoretic methods.

4.2 Marker Based Hybrid MT

Groves and Way [GW05] describe a technique which they call Marker Based EBMT and show that it has better performance than word-based SMT. The Marker Hypothesis is a universal psycholinguistic constraint which posits that languages are 'marked' for syntactic structure at surface level by a closed set of specific lexemes and morphemes.

It states that all natural languages have a closed set of specific words or morphemes which appear in a limited set of grammatical contexts and which signal that context

- In the pre-processing stage, sourcetarget aligned sentences in the parallel corpus are **segmented at** each new occurrence of a **marker word**, subject to the constraint that each marker chunk contains at least one non-marker (or content) word
- Sets of bilingual chunks are created by **aligning** based on marker tags and relative source and target chunk positions.
- The identification of cognates is implemented using the **Levenshtein distance algorithm**, with pairs of $< source, target >$ words with a distance below an empirically set threshold considered to be cognates
- **Mutual Information (MI)** scores are collected over the entire corpus

A $< source, target >$ word pair with a high MI score indicates that they co-occur frequently within the corpus

$$\mathbf{MI(x,\ y) = log2\ [P(x,\ y)\ /\ (P(x)*P(y))] = log2\ [N*f(x,\ y)\ /\ (f(x)*f(y))]}$$

- Creation of the Example Database
- Generalise Templates
- Recombination based on the following weights:

$$weight = \frac{no.\ of\ occurrences\ of\ the\ proposed\ translation}{total\ no.\ of\ translations\ produced\ for\ SL\ phrase}$$

4.3 Hybrid Rule Based and Example Based MT

Seeding with EBMT examples

Various approaches have been proposed in this area. Some approaches that use Pharaoh System and seed it with other outputs are [GW05]:

- Seeding Pharaoh with word- and phrase alignments induced via Giza++ generates better results than if EBMT sub-sentential data is used.
- Seeding Pharaoh with a hybrid dataset of Giza++ word alignments and EBMT phrases improves over the baseline phrase-based SMT system primed solely with Giza++ data.
- Seeding Pharaoh with all data induced by Giza++ and the EBMT system leads to the best performing hybrid SMT system: for EnglishFrench

Although the results obtained from the hybrid techniques proposed to date are somewhat disappointing, there are many cases where the translation obtained by merging the extracted examples with the decoder can improve the results obtained by the hybrid approach. One possible explanation is that an evaluation based on the WER metric and single translations might not fully do justice to the real contribution of the TM and hence better metrics for evaluation of MT systems are needed.

Chapter 5

Existing MT Systems and Performance

5.1 EBMT Systems

5.1.1 CMU EBMT

The system is a lexical EBMT system, meaning that it calculates similarity on the surface form of texts (Brown, 1996, 2004). In other words, given an input sentence to be translated, the system finds similar sentences in the surface form. In the system, the similarity calculation was implemented by finding contiguous source word matches in a stored example database. For each match in a sentence pair, the system finds its translation phrase using a word-to-word correspon- dence table, in which all the word-to-word mappings have a binary correspondence value indicating whether they are translations or not.

5.1.2 Marclator

Marclator is a free/open-source example-based machine translation (MT) system based on the marker hypothesis, comprising a marker-driven chunker, a collection of chunk aligners, and a simple proof-of-concept monotonic recombinator or "decoder". Marclator is largely comprised of components from MaTrEx, the data-driven machine translation system designed by the Machine Translation group at the School of Computing of Dublin City University (Stroppa and Way 2006, Stroppa et al. 2006). The MaTrEx system is a modular data-driven MT engine, built on the following established Design Patterns. It consists of a number of extendible and re- implementable modules, the most important of which are:

- Word Alignment Module: takes as its input an aligned corpus and outputs a set of word alignments.
- Chunking Module: takes in an aligned corpus and produces source and target chunks.

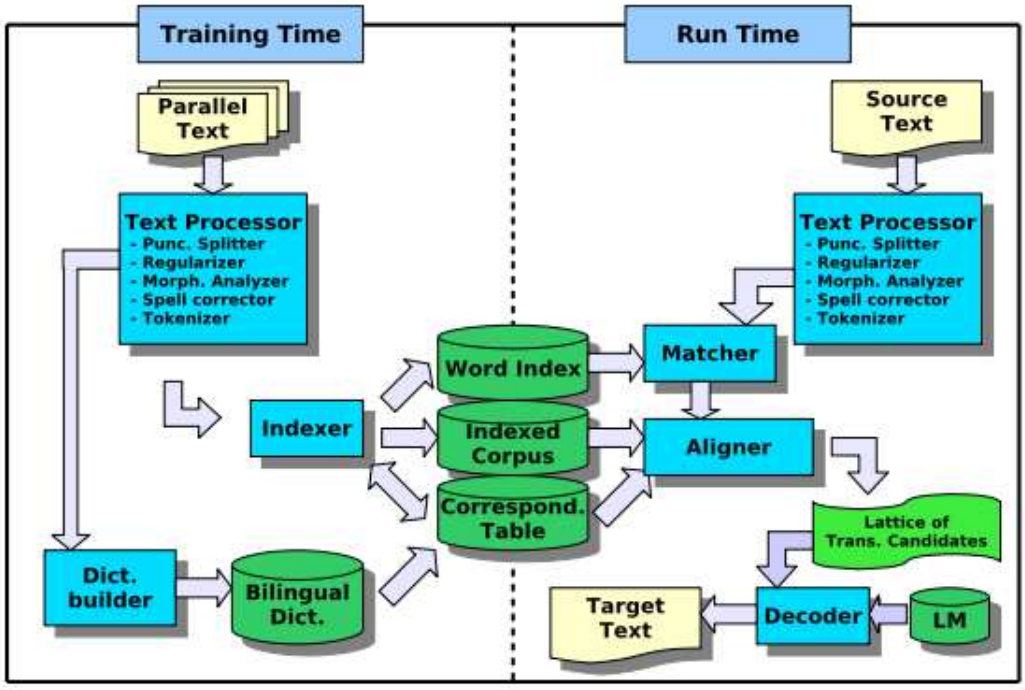


Figure 5.1: The CMU EBMT System [Kim10]

5.1.3 ALEPH - Proportional Analogies

The following gives the basic outline of how ALEPH performs the translation of an input sentence, using a given bicorpus of aligned sentences:

- Form all analogical equations with the input sentence D and with all relevant pairs of sentences (Ai , Bi) from the source part of the bicorpus ;

$$Ai : Bi :: x : D$$

- For those sentences that are solutions of the previous analogical equations and which do not belong to the bicorpus, translate them using the present method recursively. Add them with their newly generated translations to the bicorpus;
- For those sentences x = Ci,j that are solutions of the previous analogical equations and which belong to the bicorpus, do the following;
- Form all analogical equations with all possible target language sentences corresponding to the source language sentences ;

$$Ai : Bi :: Ci,j : y$$

- Output the solutions y = Di,j of the analogical equations as a translation of D, sorted by frequencies

5.1.4 Gaijin Template Driven EBMT

Example-based Machine Translation (EBMT) is a recent approach to MT that offers robustness, scalability and graceful degradation, deriving as it does its competence not from explicit linguistic models of source and target languages, but from the wealth of bilingual corpora that are now available. Gaijin (Veale & Way 1997) is such a system, employing statistical methods, string-matching, case-based reasoning and template-matching to provide a linguistics-lite EBMT solution. The only linguistics employed by Gaijin is a psycholinguistic constraintthe marker hypothesisthat is minimal, simple to apply, and arguably universal.

5.2 SMT Systems

5.2.1 GIZA++ - Aligner

GIZA++ is used to produce the alignments in parallel text. It runs the EM algorithm for each of the IBM model starting from 1 and can go up to 5. The output is the phrase table and a configuration file called moses.ini. There are various steps executed in GIZA++ when we run the train-factored-phrase-model.perl script

1. Prepare data
    - Gives IDs to each words in English and Hindi.
    - Creates the eng.vcb and hin.vcb files.
    - Converts the parallel corpora to eng_hin.snt file (as well as hin_eng.snt file).
2. Run GIZA++: This produces alignment for both direction, source-target as well as target-source
3. Align words: This Uses heuristics to combine two-way alignments. These heuristics involves Intersection,Union and Grow-Diag.
4. Get lexical translation table: Lexical probabilities are calculated for each word pair
5. Extract phrases: All phrases are dumped into a big file and they are sorted.
6. Score phrases: Counting of the phrases are done and they are assigned probabilities based on these counts. The complete entry contains source phrase, target phrases, lexical probabilities values, phrase translation probabilities in both forward as well as backward direction.
7. Build lexicalized reordering model: Distance based reordering model is built. This contains the penalty values for displacement of words from its source position.

8. Build generation models: This is used for the factored model where it is necessary to define rules for the creation of final word from its components. One of the component, is basically the lemma of the word. Other components includes the various factors like case marker, morphology etc.

9. Create configuration file: All the parameters learned during training are dumped into a configuration file called moses.ini. This contains various fields like number of iteration for each IBM Model, weight assigned to each model, language model weight, distortion weight, translation weight etc.

5.2.2 Moses Decoder

Moses is a statistical machine translation system that allows automatic training of translation models for any language pair [KHB+07]. Features

- Moses offers two types of translation models: phrase-based and tree-based
- Moses features factored translation models, which enable the integration linguistic and other information at the word level
- Moses allows the decoding of confusion networks and word lattices, enabling easy integration with ambiguous upstream tools, such as automatic speech recognizers or morphological analyzers

All Moses needs is a collection of translated texts (parallel corpus). An efficient search algorithm finds quickly the highest probability translation among the exponential number of choices. It was developed by Hieu Hoang and Philipp Koehn.It uses the phrase-table and moses.ini file generated from GIZA++ to translate a sentence in source language to target language. Moses can be used for both phrase-based as well as tree-based models. It has the support for factored translation models which enable the integration of linguistic features into statistics. It mainly uses the beam-search algorithm for pruning off the bad translation option and returns user with translation with highest probability.

5.3 Hybrid MT Systems

5.3.1 Cunie System

Cunei (R. Brown, 2009) is a hybrid platform for machine translation that draws upon the depth of research in Example-Based MT (EBMT) and Statistical MT (SMT). In particular, Cunei uses a data-driven approach that extends upon the basic thesis of EBMT–that some examples in the training data are of higher quality or are more relevant than others. Yet, it does so in a statistical manner, embracing much of the modeling pioneered by SMT, allowing for efficient optimization. Instead of using a

static model for each phrase-pair, at run-time Cunei models each instance of translation in the corpus with respect to the input. Ultimately, this approach provides a more consistent model and a more flexible framework for integration of novel run-time features.

Aaron B. Phillips. "Cunei Machine Translation Platform for WMT'10" The Fifth Workshop on Statistical Machine Translation, Uppsala, Sweden, July 2010.

Aaron B. Phillips and Ralf D. Brown. "Cunei Machine Translation Platform: System Description." 3rd Workshop on Example-Based Machine Translation, Dublin, Ireland, November 2009.

5.3.2 OpenMaTrEx System

OpenMaTrEx (Michael Forcada, Andy Way, 2010) is a free/open-source (FOS) example-based machine translation (EBMT) system based on the marker hypothesis. It comprises a marker-driven chunker, a collection of chunk aligners, and two engines: one based on the simple proof-of-concept monotone recombinator (previously released as Marclator, http://www.openmatrex.org/marclator/) and a Moses-based decoder (http://www.statmt
OpenMaTrEx is a FOS version of the basic components of MaTrEx, the data-driven machine translation system designed by the Machine Translation group at the School of Computing of Dublin City University (Stroppa and Way 2006, Stroppa et al. 2006).

Chapter 6

Machine Translation Evaluation Criteria

The evaluation of translation is nothing but measuring the correctness of the translated text. It is a major task in machine translation because there are many perfect translations for a given sentence. Human evaluation is best method for MT evaluation but due to lack of time, it is impractical. So we have BLEU and NIST, automatic evaluation tools which evaluate a translation using n-gram counts.

6.0.3 BLEU

BLEU (BiLingual Evaluation Understudy) [PRWZ01] is based on n-gram co-occurrence i.e. the quality of MT output is judged by comparing its n-grams with reference translations by the humans. As there are many perfect translations for a given sentence we have many reference translations and many candidate translations. Each candidate translation is evaluated with every reference translation and the candidate producing the best BLEU score is considered as the best of all. An extension to this method is modified n-gram precision where each reference word is considered exhausted after it is matched with any candidate word.

However even though it is a very familiar metric for MT evaluation, it has been proved that BLEU is not appropriate for Hindi-English MT (A. Ramanathan et al., 2007). It is only useful when we wish to compare translations from two different MT systems or by same systems at different points of time. BLEU fails in case of indicative translation. Indicative means rough and draft-quality translations. Indicative translations are understandable but often not very fluent in the target language. This evaluation metric considers synonyms as two different words which decrease the BLEU score. A grammatically incorrect sentence is given the same BLEU score as compared to a sentence which is grammatically correct.

In BLEU, each MT output line is ranked by a weighted average of the number of N-gram overlaps with the human translation[PRWZ02]. BLEU has the following fea-

tures:

- Modified unigram precision : To compute this, BLEU rst counts the maximum number of times a word occurs in any single reference translation. Next, it clips the total count of each candidate word by its maximum reference count , adds these clipped counts up, and divides it by the total (unclipped) number of candidate.

$$p_n = \frac{\sum_{C\epsilon\{Candidates\}} \sum_{n-gram\epsilon C} Count_{clip}(n-gram)}{\sum_{C'\epsilon\{Candidates\}} \sum_{n-gram\epsilon C'} Count_{clip}(n-gram')}$$

  This prevents BELU from calculating extra repeated words.

- BLEU uses unigram, bigrams, trigrams, and often quadrigrams. It combines the modied N-gram precisions using weighted average of the logarithm of modified precisions. This Accounts for the exponential decay of the N-gram precisions. Taking logarithm equalizes the importance of all N-grams, making them linear. It prevents the domination of unigrams and boosts up the effect of higher-grams.

- Sentence brevity penalty : BLEU considers the length of the candidate sentence and the reference sentence using a sigmoid filter. If the length of reference is equal to the length of the candidate sentence, then penalty is 0. If they differ by a small value (1-2 words more or less) , then also the penalty is not so high. But if they vary significantly then penalty increases rapidly.

6.0.4 NIST

It is another evaluation metric from US National Institute of Standards and Technology. sIt is a variation of BLEU where along with counting n-grams, the information content of a particular n-gram is also calculated. The rarer is the n-gram the more weight is assign to it. For combining scores from different N-grams, NIST suggest using Arithmetic mean instead of Geometric mean as in BLEU. NIST gives higher weights to more informative N-Grams *i.e*, less frequent N-Grams. It uses information gain rather than precision. For example: The correctly matched trigram such as *is called as* is assigned a lesser weight in comparison to the correctly matched trigram *Electrical and Electronic*.

6.0.5 Meteor

The Meteor automatic evaluation metric scores machine translation hypotheses by aligning them to one or more reference translations. Alignments are based on exact, stem, synonym, and paraphrase matches between words and phrases. Segment and system level metric scores are calculated based on the alignments between hypothesis-reference pairs. The metric includes several free parameters that are tuned to emulate various human judgment tasks including WMT ranking and NIST adequacy. The

current version also includes a tuning configuration for use with MERT and MIRA. Meteor has extended support (paraphrase matching and tuned parameters) for the following languages: English, Czech, German, French, Spanish, and Arabic. Meteor is implemented in pure Java and requires no installation or dependencies to score MT output. On average, hypotheses are scored at a rate of 500 segments per second per CPU core.

6.0.6 Subjective Evaluation

This involves translations being judged by human evaluators for adequacy and fluency. Trans- lations are being asigned score on the scale of 1 to 5 for both these criteria according to the following table [RBH+08].

Level	**Interpretation**
5	Flawless Hindi, with no grammatical errors whatsoever
4	Good Hindi, with a few minor errors in morphology
3	Non-native Hindi, with possibly a few minor grammatical errors
2	Disfluent Hindi, with most phrases correct, but ungrammatical overall
1	Incomprehensible

Table 6.1: Subjective Evaluation - Fluency Scale

Level	**Interpretation**
5	All meaning is conveyed
4	Most of the meaning is conveyed
3	Much of the meaning is conveyed
2	Little meaning is conveyed
1	None of the meaning is conveyed

Table 6.2: Subjective Evaluation - Adequacy Scale

Chapter 7

Summary

Through the course of this survey, a broad background for Example-Based Machine Translation was established. Where it is placed in the MT Taxonomy, how it relates to Translation Memory and comparison of EBMT with Rule Based and Statistical Modeling Techniques was studied. Various matching techniques that are used in example-based translation were understood.

It was learnt that EBMT essentially takes a stance between knowledge based and statistical machine translation systems. Many example-based machine translation systems combine both rule-based and corpus driven techniques.

- The linguistic knowledge of the system can be easily enriched in case of EBMT by simply adding more examples as opposed to knowledge-driven approaches where rules need to be stated explicitly.
- EBMT systems are corpus-driven, so they cover examples/constructions that really occur and ignore the ones that do not, thus reducing overgeneration.
- EBMT has the potential to combine both data driven and rule based approaches to MT. Hence hybrid systems in the example-based domain seem promising.

In general, EBMT tackles the following problem: given the translation for the fragments of a source utterance, including its words, phrases and other non-constituent chunks, infer the best choice of its translation in the target language with respect to the available translated fragments. There must be at least a sequence of fragments that covers the entire input utterance; otherwise, the input cannot be trans- lated completely. To enhance the capability of translation, it is necessary to collect translated fragments from existing parallel corpora, via text alignment and example acquisition.

This survey also covered a description of the Statistical approach to machine translation. The hybrid techniques were discussed briefly followed by a description of the existing MT systems and Machine Translation evaluation criteria.

Bibliography

[BCP+90] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 1990.

[Bha08] Pushpak Bhattacharyya. Machine translation, language divergence and lexical resources, 2008.

[BPPM93] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, pages 263–311, 1993.

[Car99] Michael Carl. Inducing translation templates for example-based machine translation. *MTSummit VII*, 1999.

[CG01] Ilyas Cicekli and H. Altay Guvenir. Learning translations templates from bilingual translation examples. *Applied Intelligence*, 2001.

[CPP94] Lambros Cranias, Harris Papageorgiou, and Stelios Piperidis. A matching technique in example-based machine translation. *Coling: the 15th International Conference on Computational Linguistics*, 1994.

[CW03] Michael Carl and Andy Way. *Recent Advances in Example-Based Machine Translation*. Kluwer Academic Publishers, 2003.

[DPB02] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. Interlingua based english hindi machine translation and language divergence. *Journal of Machine Translation*, 17, 2002.

[DSMN10] Sandipan Dandapat, Harold Somers, Sara Morriessy, and Sudip Kumar Naskar. Mitigating problems in analogy-based ebmt with smt and vice versa: a case study with named entity transliteration. *PACLIC 24*, November 2010.

[GW05] D. Groves and Andy Way. Hybrid example-based smt: the best of both worlds? *Proceedings of the ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, 2005.

[HS92] J. Hutchins and H. Somers. *An introduction to Machine Translation.* Academic Press, 1992.

[JM08] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition).* Prentice Hall, 2008.

[KBC10] J. D. Kim, Ralf D. Brown, and J. G. Carbonell. Chunk-based ebmt. *EAMT*, 2010.

[KHB+07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session*, 2007.

[Kim10] Jae Dong Kim. *Chunk alignment for Corpus-Based Machine Translation.* PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2010.

[KOM03] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase based translation. *In Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2003.

[Nag84] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and Human Intelligence*, pages 173–180, 1984.

[NDG93] Sergei Nirenburg, Constantine Domashnev, and Dean J. Grannes. Two approaches to matching in example-based machine translation. *TMI: The Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, 1993.

[PF99] E. Planas and O. Furuse. Formalizing translation memories. *Machine Translation Summit VII*, pages 331–339, 1999.

[Phi07] Aaron B. Philips. Sub-phrasal matching and structural templates in example-based mt. *Theoretical and Methodological Issues in Machine Translation*, 2007.

[PRWZ01] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. *IBM Research Report, Thomas J. Watson Research Center*, 2001.

[PRWZ02] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[RBH+08] Ananthakrishnan Ramanathan, Pushpak Bhattacharyya, Jayprasad Hegde, Ritesh M.Shah, and M. Sasikumar. Simple syntactic and morphological processing can help english-hindi statistical machine translation,. *Proceedings of IJCNLP*, 2008.

[SD04] Harold Somers and Gabriela Fernandez Diaz. Translation memory vs. example-based mt: What is the difference? *International Journal of Translation*, 16.2, 2004.

[SDN09] Harold Somers, Sandipan Dandapat, and Sudip Kumar Naskar. A review of ebmt using proportional analogies. *EBMT 2009 - 3rd Workshop on Example Based Machine Translation*, 2009.

[SIK90] Eiichiro Sumita, Hitoshi Iida, and Hideo Kohyama. Translating with examples: a new approach to machine translation. *Third international conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, pages pp.203–212, 1990.

[SN90] Satoshi Sato and Makoto Nagao. Toward memory-based translation. *Proceedings of the 13th conference on Computational linguistics*, 3:247–252, 1990.

[Som99] Harold Somers. Review article: Example based machine translation. *Machine Translation, Volume 14, Number 2, pp. 113-157(45).*, 1999.

[Sum01] Eiichiro Sumita. Example-based machine translation using dp-matching between word sequences. *ACL-EACL Workshop Data-driven machine translation*, 2001.

[Vau76] B. Vauquois. Automatic translation - a survey of different approaches. *Statistical Methods in Linguistics*, 1976.