

# A Morpho - Syntax Based Adaptation and Retrieval Scheme for English to Hindi EBMT

Deepa Gupta

Niladri Chatterjee

Department of Mathematics

I.I.T Delhi, Hauz Khas

New Delhi, INDIA -110016

{gdeepa, niladri}@maths.iitd.ac.in

## Abstract

This paper focuses on Example Based Machine Translation (EBMT) between English and Hindi, the most popular language in South Asia. Given an input sentence, an EBMT system retrieves similar sentence(s) from its example base and adapts their translation(s) suitably to generate the translation of the given input. This paper proposes a systematic adaptation scheme that takes into account the morphology and syntax of the input and the retrieved source language sentences. The advantage of this method is that it provides an objective way of measuring the adaptation cost, and therefore can be used as a good yardstick to measure the similarity between two sentences. The proposed scheme has been elaborated with examples, the technique for estimating adaptation cost has been demonstrated. This paper also illustrates the superiority of this scheme over some existing similarity measurement schemes.

**Key Words** – Morpho-syntactic tags, Sentence patterns, Adaptation, Similarity.

## 1 Introduction

An Example-Based Machine Translation (EBMT) (Nirenburg., 1993; Sato., 1992) system uses its repertoire of past translation examples to generate the translation of a given input sentence.

Two key operations pertaining to EBMT are:

1. Retrieval – i.e. selecting an appropriate sentence from the system's database that is similar to the given input sentence;
2. Adaptation – i.e. carrying out necessary modification in the retrieved example to suit the requirement of the current input.

Evidently, the success of an EBMT depends significantly on the efficiency of its adaptation scheme.

Here we present an adaptation scheme for English to Hindi translation using the *morpho-syntactic* tags of the constituent words of the input and the retrieved sentences. The morpho-syntactic tag of a word indicates its syntactic function in the sentence. The tags are helpful in identifying the root words, their roles in the sentence and roles of the different suffixes (used for declensions) in the overall sentence construction. Fig. 1 provides an example of the records stored in our example base. A record contains the input sentence, its Hindi translation, the root word correspondence and also the *morpho-syntactic* tags of the words obtained by using the tagging scheme proposed in <http://www.lingsoft.fi/cgi-bin/engcg> for English sentences. Fig. 1 also illustrates the role of suffixes in English (e.g. “*ing*” to the root verb to derive its present continuous form, and “*s*” for plural number). Similarly, in Hindi too one may use suffixes for declension. Section 2 discusses Hindi suffixes in detail.

The adaptation scheme proposed in this paper generates the desired translation by modifying the root words and/or the suffixes in accordance with the grammars of the source and

**English sentence:** The horses have been running for one hour.  
**Tagged form:** @DN> DET CENTRAL ART SG/PL "the", @SUBJ N NOM PL "horse" %ghodaa%, @+FAUXV V PRES -SG3 "have", @-FAUXV PCP2 "be", @-FMAINV PCP1 "run" %daudaa%, @ADVL PREP "for" %, @QN> NUM CARD "one" %ek%, @<P N NOM SG "hour" %ghantaa%.  
**Hindi sentence:** # ghode ek ghante se daudaa rahen hain #

**Figure 1. An Example Sentence and its Morpho-Syntactic Tags**

the target language. It is a rule-driven approach that considers the discrepancy between the input and the retrieved sentence in the source language. The rules are formed by taking into account the grammars of both the source and the target languages. The rules help in a systematic step-by-step modification of a retrieved translation example (consisting of an English sentence and its Hindi translation) to the desired translation. The scheme has the advantage that it can estimate the total computational cost in adapting a particular retrieved example into the desired translation. This *a priori* estimate of adaptation cost may be used in designing an effective retrieval scheme that adds to the efficiency of the EBMT system.

The paper is organised as follows. Section 2 gives an overview of suffixes in Hindi. Section 3 and 4 discuss the proposed adaptation procedure and the cost estimation (for adaptation) scheme, respectively. Section 5 compares the proposed approach with existing similarity measurement schemes.

## 2 An Overview of Suffixes in Hindi

Some examples of usage of Hindi suffixes for declension are given below:

*To change the Number for Nouns.* There are six possible suffixes for singular to plural conversion in Hindi (Kellogg and Grahame., 1965 ): ‘en’, ‘yaan’, ‘iyaan’, ‘an’, ‘yen’, and ‘e’ For example:

Singular		Plural
<i>chidiyaa</i>	(bird)	<i>chidiyaan</i>
<i>ghodaa</i>	(horse)	<i>ghode</i>
<i>kakshaa</i>	(room)	<i>kakshayen</i>

*Declensions of Inflected Nouns.* There are some rules for making inflected nouns. Some of them are as follows:

1. Masculine singular nouns ending in “aa” change into “e” when some case ending is added : e.g. *ladkaa* + *ne* ~ *ladke ne*. Nouns ending in other vowels do not undergo such changes ( e.g. *ghar ko*, *daaku kaa*).
2. If a noun (masculine or feminine) ends in “a”, it is changed into “aon” in plural, when a case ending is added. For example: “in the house” ~ “*ghar main*” while “in the houses” ~ “*gharon main*”. Note that, normally the plural of “*ghar*” is “*ghar*”. But because of the case ending it changes to “*gharon*” in the above example.

*To modify the Adjective.* Adjectives in Hindi are modified according to the gender and number of the corresponding noun. Some of the rules are:

1. If an adjective in Hindi ends in “aa” it changes into “e” for plural. E.g. *achchhaa ladkaa* (good boy) and *achchhe ladke* (good boys).
2. An adjective ending with “aa” changes into “ii” for feminine. E.g. *achchhii ladkii* (good girl) and *achchhii ladkiyaan* (good girls).

*Verb Morphology.* Morphology of verbs in Hindi depends upon the gender, number and person of the Subject. There are 11 possible suffixes (e.g. *taa*, *tii*, *egaa*) in Hindi that may be attached to the root Verb. Also some auxiliary verbs (e.g. *hai*, *hain*,) are used. For example:

He reads.	→	<i>wah padtaa hai</i>
She reads.	→	<i>wah padtii hai</i>
He will read.	→	<i>wah padegaa.</i>

Section 4 discusses how auxiliary verbs are taken care of in adaptation. Section 3 discusses operations involving the words and suffixes in a retrieved example for suitable adaptation.

### 3 Adaptation Procedure using Word and Suffix Operations

The proposed adaptation scheme is based on seven different operations:

*Word Replacement (WR)*: Each WR operation replaces one word of the retrieved example with a suitable word. If the input sentence is: “Ram is eating rice”, and the retrieved example is “Ram is eating bread ~ *ram rotii khaa rahaa hai*”, then to generate the translation, one just needs to replace “bread (*rotii*)” with “rice (*chawal*)”.

*Word Deletion (WD)*: Through this operation some words of the retrieved example are deleted. For illustration, suppose the input sentence is “Animals were dying of thirst”, and the retrieved translation example is “Birds and animals were dying of thirst ~ *pakshii aur pashu pyassa se mar rahii thii*”. The desired translation can then be obtained by deleting the “birds and (*pakshii aur*)” part from the retrieved translation.

*Word Addition (WA)*: Each WA operation suggests addition of a new word to the retrieved translation example. For illustration, one may consider the example given just above with the roles of input and retrieved sentences reversed.

*Suffix Addition (SA)*: Here a suffix is added to some word in the retrieved example. Note that, the word here is in its root form.

*Suffix Deletion (SD)*: By this operation the suffix attached to a word may be removed and the root word may be obtained.

*Suffix Replacement (SR)*: Here a suffix in a word is replaced with a different suffix to meet the current translation requirements.

*Copy (CP)*: When a word or suffix of the example is retained intact in the new translation then we call it copy operation.

Fig. 2 provides an example of adaptation using the above operations. In this example the input sentence is "Sita sings ghazals well", and the retrieved translation example is: "He is singing ghazal ~ *wah ghazal gaa rahaa hai*". The

translation to be generated is : "*sita ghazalen achchhii gaatii hai*". When carried out the adaptation using both word and suffix operations the adaptation steps look as follows:

<b>Input</b>	<i>wah</i>	<i>ghazal</i>	<i>gaa</i>	<i>rahaa</i>	<i>hai.</i>
	↑	↓	↓	↑	↓
<b>Operation</b>	WR	SA	SA	WD	CP
	↓	↓	↓	↓	↓
<b>Output</b>	<i>sita</i>	<i>ghazalen</i>	<i>gaatii</i>	ϕ	<i>hai</i>

**Figure 2. Example of Adaptation**

Note that if the retrieved example is “Ram is playing cricket ~ *ram cricket khel rahaa hai*” then also one may get the desired output but with more number of operations. However if the retrieved example is completely different from the input sentence (e.g. “Can sita sing some songs today ~ *kyaa aaj sita kuchh gaane gaa saktii hai*”) then its adaptation will be computationally even more expensive and will involve more complicated reasoning.

The above discussion suggests that variety of examples may be adapted to generate the desired translation, but with varying computational costs. For efficient performance an EBMT system therefore needs to retrieve an example that can be adapted to the desired translation with least cost. This brings in the notion of “similarity” among sentences. The proposed adaptation procedure has the advantage that it can provide a systematic way of evaluating the overall adaptation cost. This estimated cost may then be used as a good measure of similarity for appropriate retrieval from the example base. Section 4 discusses how the costs for the proposed adaptation method may be estimated.

### 4 Cost Evaluation for Adaptation Based on Word and Suffix Operations

The cost of adaptation depends on the number of operations required for adapting a retrieved example. Total cost is the sum of individual cost of each operation used for the adaptation. Further, to carry out the above operations the system should have access to an English to Hindi dictionary; and several operations (such as WA, WR) require a search through the dictionary. Cost measurement scheme therefore should take into account the following:

- Although word operations involve dictionary search, suffix operations involve only the relevant suffixes in the languages concerned. Since the number of suffixes is limited, their use reduce dictionary search significantly.
- Since sentence structures in a language are guided by strict syntactic rules, it is straightforward to formulate rules regarding operations on suffixes in a given context.
- For Word Deletion, in order to avoid computationally expensive dictionary search, one may store several information (as given in Fig. 1) in an example record.

The following points may be noted regarding our implementation of the proposed scheme:

- 1) Total cost of each operation depends on the *search time* and the number of steps executed. This number is called *step counts* (Horwitz et al., 2000).
- 2) Average Search Time is being used to measure the complexity of the dictionary search procedure. Currently we are using Sequential Search. However, one may use other search algorithms (such as, binary search), but that does not affect the relative cost of the word and suffix operations.
- 3) In order to reduce the search time, instead of using one dictionary, we are using different word databases for different POS. Our word databases (courtesy *Sabdanjali* dictionary of IIT Hyderabad) are of the following sizes: Noun—13953, Adjective— 5449, Adverb—1027, Preposition-87, Pronoun-72 and Verb—4330.
- 4) For applying any word or suffix operation, one needs to first find the appropriate word position in a retrieved example. If the sentence length is  $L$ , the average search time is  $\propto L/2$ . However, in cases where the position is already prescribed by the syntax of the language, one may directly access the right position. In such cases the search time is considered to be  $\propto 1$ .
- 5) Since the number of suffixes is fixed, we assume fixed costs ( $K$ ) for all the suffix operations.

Section 4.1 describes how the computational cost of each of the adaptation operations is computed in view of the above assumptions.

#### 4.1 Cost of Different Adaptation Operations

The cost of the seven different operations are estimated in the following way:

*Word Deletion:* To delete a word from a retrieved example, first the word is located in the sentence, and then it is deleted. Thus the average cost is  $c * L/2 + \epsilon_1$ , where  $c$  is the constant of proportionality, and  $\epsilon_1$  is a small positive quantity reflecting the cost of actual deletion operation (e.g. adjustment of pointers if sentences are stored in a linked-list structure of words).

*Word Addition:* Word addition is done in three steps. First, the Hindi equivalent of the word to be added is found in the dictionary. Then the position (in the sentence) where the new word has to be added is located. Finally, the actual addition is done. Average time requirement for a WA operation is therefore  $d * D/2 + c * L/2 + \epsilon_2$ . Here  $\epsilon_2 (> \epsilon_1)$  is a small number indicating the cost of adding the new word in the retrieved translation. Here  $d$  is the constant of proportionality for retrieval from the dictionary; and  $c$  and  $L$  are as given above. If the dictionary is in an external storage then  $d$  will be different (in fact,  $d \gg c$ ) from  $c$ . However, if the dictionary is copied into the RAM of the machine it may be assumed to be same as  $c$ .

*Word Replacement:* The activities here are similar to what needs to be done in WA, except that here no space is required to be created for the new word. The cost is therefore reduced by  $\epsilon_2$ . Hence the average cost is  $d * D/2 + c * L/2$ .

*Suffix Deletion:* This operation is beneficial when the root word is same in both the input and the retrieved English sentences. Here the work involved is first to identify the right suffix, then to do the stripping. So the cost is  $c * (L/2) + \theta$ , where  $\theta$  is a very small quantity reflecting the cost of identifying the suffix and its stripping.

*Suffix Addition :* Suffix addition is done in two steps. First the position of the word where the suffix has to be added is determined. The average cost for this operation is  $c * L/2$  (as explained above). Next the suffix database is searched for obtaining the appropriate suffix.

The average cost therefore is:  $K + c*(L/2)$ , where  $K$  is as explained in Section 4 above.

*Suffix Replacement* : In a similar manner, here the cost is  $K + c*(L/2) + \theta$ . This operation is costlier than SA because here on the top of adding the suffix some extra computational effort is spent in identifying the suffix to be replaced and then in its stripping from the word.

For *Copy* operation no computational cost is taken into account. In Section 4.2 we now discuss how cost may be calculated for adaptation between different sentence structures.

#### 4.2 Cost due to Types of Sentence Structure

For both English and Hindi the structure of sentences varies with different features, such as,

1. *Type of sentence*. Whether the sentence is affirmative, negative, interrogative etc.
2. *Tense and Form of the Verb*. Since there are three tenses (i.e. Present, Past and Future) and four forms (Indefinite, Continuous, Perfect, and Perfect Continuous), in all one can have 12 different structures.
3. *Variations in Subject and Object*. These variations may happen in many different ways, such as, Proper Noun, Common Noun (Singular or Plural), Pronoun, Verb (Infinitive or Gerund), and Possessive Case .

Similarly, the Voice of the sentence (Active or Passive), Modals (such as shall, should, may, might, ought to) impose specific structural types in Hindi. Systematic study of these patterns helps in estimating the adaptation costs between them. Due to lack of space we elaborate variations due to Kind of Sentence and Tense (and Form) of Verbs only.

#### Costs due to Variations in Kind of Sentences

Here we consider four kinds of sentences: Affirmative (AFF), Interrogative (INT), Negative (NEG), and Negative-Interrogative (NINT). Typical sentence structures of these four types are given in Figure 3.

Ram eats rice. ~ <i>ram chawal khaataa hai.</i>
Ram does not eat rice. ~ <i>ram chawal naheeng khaataa hai.</i>
Does Ram eat rice? ~ <i>kyaa ram chawal khaataa hai?</i>
Does Ram not eat rice? ~ <i>kyaa ram chawal naheeng khaataa hai</i>

**Figure 3. Some Typical Sentence Structures**

One may notice that In Hindi the negative and interrogative structures are obtained by addition of the words “*naheeng*” and “*kyaa*”. Also note that the position of “*kyaa*” is always at the beginning of the sentence – hence its addition or deletion needs no traversing through the sentence. The costs of these operations are therefore very negligible. By referring to the notations given in Section 4, we denote the cost of WA (for “*naheeng*”) as  $k1 \cong L/2 + \epsilon_2$ , cost of WD (for “*naheeng*”) as  $k2 \cong L/2 + \epsilon_1$ , cost of WA (for “*kyaa*”) as  $k3 \cong \epsilon_2$  and cost of WD (of “*kyaa*”) as  $k4 \cong \epsilon_1$ . Table 1 gives the cost of all types of variation from input to retrieved sentences. The expressions are obtained by deciding upon which of the words are being added and/or deleted for the adaptation.

Input Ret'd	AFF	NEG	INT	NINT
AFF	0	$k1$	$k3$	$k1 + k2$
NEG	$k2$	0	$k3 + k2$	$k3$
INT	$k4$	$k1 + k4$	0	$k1$
NINT	$k2 + k4$	$k4$	$k2$	0

**Table 1. Cost due to Variation in Kind of Sentences**

#### Cost due to Verb Morphological Variation

Hindi verb morphological variations depend on four aspects: *tense* (and *form*) of the sentence, *gender*, *number* and *person* of subject. All these variations affect the adaptation procedure. In Hindi, these conjugations are realized by using *suffixes* attached to the root verbs, and/or by adding some auxiliary verbs. We call them "Morpho-Words" (MW). Below we illustrate how MWs can be used in cost estimation.

Input Ret'd	M1	F1	N1 (N3)	M2	F2	M3	F3
M1	0	$s_1 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$
F1	$s_1 + L/2$	0	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$
N1 (N3)	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	0	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$
M2	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$	0	$s_1 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$
F2	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + L/2$	0	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$
M3	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	0	$s_1 + L/2$
F3	$s_1 + s_2 + L/2 + \gamma$	$s_2 + L/2$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + s_2 + L/2 + \gamma$	$s_1 + L/2$	0

**Table 2. Cost of verb morphology for Present Continuous to Present Continuous**

Consider the input sentence “He is eating rice”. The desired translation is “*wah chawal khaa raha hai*.” Suppose also that the retrieved example is. “We are eating rice ~ *ham chawal khaa rahen hai*”. The right translation is obtained by two word replacements in the verb morphology of the retrieved translation: “*rahen*” by “*rahaa*” and “*hai*” by “*hai*”. (The need to replace the subject ‘*ham*’ with ‘*wah*’ is not part of the present discussion).

Since there are 12 different structures depending upon the tense and form, in all one may have rules for all the 12 x 12 many transformations.

Table 2 explains the costs due to Verb Morphology considering different possibilities of subjects in case of Present Continuous to Present Continuous, where the column and row headers indicate the person and gender of the subject of the input and retrieved sentence. For instance, M1, F2 and N3 represent 1<sup>st</sup> person masculine, 2<sup>nd</sup> person feminine, 3<sup>rd</sup> person neuter respectively. However, the column and row corresponding to N3 are not required in case of Hindi as the treatment of 3<sup>rd</sup> and 1<sup>st</sup> person neuter gender are same. Hence the column and row corresponding to N1 will be used for both N1 and N3. Similar tables can be made for

transformations between all the different pairs of verb morphology.

In general, a transformation from Present Continuous to Present Continuous requires at most two Word Replacements:

1. Replacement of the MW of the form {*rahaa, rahen, rahii*} by one from the same group. The average cost ( $s_1$ ) for which is 3/2.
2. Replacement of the MW of the form {*hai, ho, hoon*} by one from the same set. Here the average cost ( $s_2$ ) is 4/2 = 2.

Note that if the person and the gender of a subject in both input and retrieved sentences are same then the cost of replacement for MW is nil. In cases where word replacements are required, the total cost is the sum of the three following components:

1. The cost of searching the position where the replacement is carried out. Here the average cost is proportional to the length of average Hindi sentence i.e. L/2. Even if two word replacements are necessary, one search is sufficient for locating both as the words occur in consecution.
2. The cost for the morphological transformation which may be  $s_1$ ,  $s_2$  or  $s_1 + s_2$  depending upon the case.

- Some additional cost  $\gamma$ , where  $\gamma$  is a very small positive number.

With respect to the example given above the cost of adaptation due to Verb Morphology is  $s_1+s_2 +L/2 + \gamma = 2 +1.5 +2.5 +\gamma = 5 +\gamma$ , by referring to the cell (M3, N1).

### 4.3 Use of Adaptation Cost as a Measure of Similarity

The estimated cost of adaptation (with respect to a particular database and the underlying search procedure) may be used as an *a priori* measurement of similarity for effective retrieval. The input sentence may be compared with the example base sentences in terms of *morpho-syntactic* tags, their discrepancies may be measured, and adaptation cost may be estimated using the formulae given above. The example base sentence having the minimum cost of adaptation may then be considered as most similar to the input sentence. We have applied this technique on an example base of 1000 sentences and the results are given in the following section.

## 5 The Proposed Approach vis-à-vis Some Similarity Measurement Schemes

We are comparing our scheme with two methods of similarity measurement given in (Manning and Schutze, 1999.). These methods are as follows:

The first method is based on *semantic similarity*. Here similarity is measured on the basis of commonality of words occurring in the input sentence and each sentence of the database. The input and each database sentence are represented in a high-dimensional vector space. Each dimension of the space corresponds to a distinct word in the database. The similarity is then calculated as the dot product of the vectors. Table 3 gives the results when the proposed algorithm has been applied to select the best match for the input sentence: "*Sita sings ghazals*" from the given example base of 1000 sentences.

The main drawback of this algorithm is that the outcome varies significantly on the content words and the size of the database sentences, and the occurrence of the words in the sentences.

Example Sentence	Semantic Score
Sita sings ghazals.	1.00
He has been singing ghazals	0.175
Sita is singing a melodious song.	0.033
Sita is eating rice	0.033
Sita is going home by car	0.033

**Table 3. Semantic Similarity Values**

In the second method the measurement is done on the basis of syntax. This needs all the sentences to be tagged at *morpho-syntactic* level. Here, too, similarity is measured in terms of dot products of vectors. The vectors are formed using the morpho-syntactic tags of the constituent words. When the vector-based technique was applied for the same input sentence "*Sita sings ghazals*", the sentences given Table 4 are retrieved as the best five matches. Note that here similarity of words is completely ignored, as the main emphasis is laid on the similarity of tense.

Example sentences	Syntactic Score
Sita sings ghazals.	1.000
Sita reads history.	0.999304
He reads history.	0.993120
Babies drink milk	0.975850
She eats mangoes	0.918291

**Table 4. Syntactic Similarity Values**

Table 5 gives the best five matches when the retrieval is made by the scheme proposed in the paper using the same input sentence and the same example base. Cost here is measured according to scheme given in Section 4. The results clearly show the superiority of the proposed algorithm over the technique discussed just above.

Example sentence	Adaptation cost
Sita sings ghazals.	0
He has been singing ghazals.	$9+\epsilon_1$
Sita sang ghazal.	10
Sita is singing melodious song.	$6996+2\epsilon_1$
Sita reads history.	9147.5

**Table 5. Retrieval on the basis of cost of word and suffix operation**

## 6 Conclusion

The present work considers English to Hindi EBMT. Since success of EBMT depends heavily upon the retrieval scheme, the more similar is the retrieved example sentence to the input one, the easier is its *adaptation* to the present translation requirement, and consequently generation of the required translation will be more cost-effective. However, no significant scheme has so far been developed to quantify the similarity between two sentences in a systematic way. The primary difficulty here is that there is no unique way of defining similarity. As a consequence, different approaches for measuring similarity may be found in literature: word-based metrics (Nirenburg., 1993), syntax-rule driven metrics (Sumita and Tsutsumi., 1988), character-based metrics ( Sato., 1992), linear-regression model (Chatterjee., 2001), as well as some hybrid methods.

The present work makes an extensive and in depth study of a retrieval scheme based on the morphology and syntax of sentences. Various adaptation operations involving words and suffixes have been proposed and estimation of costs for these operations have been formulated. Since these adaptation techniques involve lexicon search, the costs have been estimated on the basis of average search time from the lexicon and total step counts.

We have applied this technique for English to Hindi Example Based Machine Translation. Since Hindi is structurally similar with many other Indian languages, the same approach may be extended to different languages of the subcontinent as well. Experiments on an example base of 1000 sentence shows that the proposed technique provides results that are qualitatively better than some existing techniques.

A limitation of the work done so far is that it considers only simple sentences. A natural extension will be to deal with more complicated sentence structures. We are currently working on the extension of the algorithm for different types of sentence structures and also for complex sentences. A complex sentence consists of one Main Clause and one or more Subordinate Clauses. Subordinate clauses may be of three types: Noun Clause, Adjective Clause and Adverb Clause (Wren et. al., 1989). However,

none of parsers that we have checked so far provides clause information of a sentence. We are therefore planning to identify clauses on the basis of connectives (subordinate conjunctions). There are specific connectives for different types of clauses. For example, there are 12 connectives for a noun clause (e.g. *who, whom, when, where*). Similarly there are specific connectives for Adjective clause and Adverbial Clause. The difficulty here is that the same connective may have different roles in different sentences. Hence dealing with complex sentences needs schemes for identifying the clause types based on the connectives and the clauses themselves. We are currently working towards this direction.

## References

- Chatterjee, N. 2001. A Statistical Approach to Similarity Measurement for EBMT. *Proc. STRANS-2001*, IIT Kanpur, 122-131.
- Horowitz, E., S. Sahni and S. Rajasekaran. 2000. *Fundamentals of Computer Algorithms*, Galgotia Publications Pvt. Ltd., New Delhi.
- Kellogg, Rev. S.H., and B. T. Grahame. 1965. *A Grammar of the Hindi Language*. Routledge & Kegan Paul Ltd, London.
- Manning, C.D. and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing* The MIT Press, MA.
- Nirenburg, S. 1993 Two Approaches of Matching in Example-Based Machine Translation, *Proc. TMI-93*, Kyoto, Japan.
- Sato, S. 1992. CTM: An Example-Based Translation Aid System. *Proc. Of COLING*, 1259-1263.
- Sumita, E. and Y. Tsutsumi. 1988. A Translation Aid System Using Flexible Text Retrieval Based on Syntax Matching. *TRL Research Report*, Tokyo Research Laboratory, IBM.
- Wren, P.C., H. Martin and N.D.V.P. Rao. 1989. *High School English Grammar*. S.Chand & Co. Ltd., New Delhi.