
English to Arabic Example-based Machine Translation System

Assist. Prof. Suhad M. Kadhem, Yasir R. Nasir

Computer science department, University of Technology

E-mail: suhad_malalla@yahoo.com, Yasir_rmfl@yahoo.com

Received: 5/11/2014

Accepted: 19/5/2015

Abstract – **T**he Example Based Machine Translation (EBMT) system retrieves similar examples (pairs of source phrases, sentences, or texts and their translations) from a database of examples, adapting the examples to translate new input. The Example Base (EB) is an important component in an EBMT system. It handles the storage to support the translation process. Thus, an efficient EB must be capable of handling a massive volume of examples at an adequately high speed.

In this research, a new approach to reduce the redundancy problem that some EBMT systems suffer from is suggested by designing EB using B^+ tree. The EB is used to store the examples of a particular field in a manner that reduces the redundancy of these examples (or even sub examples) in order to provide efficient memory usage and to minimize the search time. The lexicon of the proposed method is represented by using two databases. One database is used for storing the English words and another database is used for storing the English transfer grammars.

Keywords: – EBMT, EB, B^+ Tree.

1.Introduction

Stepping into Information Age, language as the information carrier has become the most significant means for human to communicate. But it has been considered as the barrier of communications between people from different countries. The problem of converting a language into another quickly and efficiently has become a problem of common concern for humanity [1].

Machine Translation (MT) is an automatic translation of one language into one or more languages by means of a computer or another machine that contains a dictionary along with the programs needed to make logical choices from synonyms, supply missing words, and rearrange word order as required for the new language [2].

In this research C B⁺ tree method will be used to store the examples in the Example Base (EB) part of the (EBMT) system to reduce the redundancy of these examples and to provide efficient memory usage.

2.Approaches to MT

A machine translation system first analyses the source language input and creates an internal representation. This representation is manipulated and transferred to a form suitable for the target language. Then at last output is generated in target language. Based on the degree of dependence of internal representation on the source and target languages, MT can be classified into several approaches [3]:

2.1 Direct Machine Translation

Direct MT systems provide direct translation. No intermediate representation

or complex architecture will be involved. It carries out word by word translation with the help of a bilingual dictionary usually followed by some syntactic rearrangement. Due to this direct mapping, such systems are highly dependent on both the source and target languages [4].

2.2 Rule-Based Machine Translation

Rule-Based Machine Translation (RBMT) system consists of a collection of rules called grammar rules, a bilingual or multilingual lexicon, and software programs to process the rules. The rules play a major role in various stages of translation such as syntactic processing, semantic interpretation, and contextual processing of language [5].

In RBMT, the core process (transfer) is mediated by bilingual dictionaries and rules for transforming SL structures into TL structures and/or by dictionaries and rules for deriving (intermediary representations) from which output can be produced. The preceding stage of analysis interprets input SL strings into suitable translation unit. The succeeding stage of synthesis (generation) derives TL output text from the TL structures or representations generated by the transfer process [2].

RBMT systems parse the source text and produce an intermediate representation. Based on the intermediate representation used this approach is further classified into the following approaches [3]:

2.2.1 Transfer Based Machine Translation

Transfer based system can be broken down into three different stages: analysis, transfer and generation. In the first stage,

the source language parser is used to produce the syntactic representation of the source language sentence (Internal representation). In the next stage, the result of the first stage is converted into equivalent target language representation (another internal representation). Finally, a target language morphological analyser is used to generate the target language text [5].

2.2.2 Inter-lingua Machine Translation

In this approach, the source language is analyzed and then converted into a single internal representation that is independent of both the source and the target languages called “Interlingua” from which translations can be generated to different target languages. In short, the translation in this approach is a two-stage process, i.e. analysis and synthesis [1].

2.3 Corpus-Based Machine Translation

This approach uses a large amount of raw data in the form of parallel corpora. This raw data contains texts, dictionaries, grammars, etc. and their translations. These corpora are used for acquiring translation knowledge [3]. In recent years there is an increased interest in corpus based MT systems, because it needs less effort from the language/linguistic experts and less human effort is required. Corpus based approach is further classified into the following types [4]:

2.3.1 Statistical Machine Translation

SMT is a method for translating text from one natural language to another based on the knowledge and statistical models extracted from bilingual corpora. A supervised or unsupervised statistical machine learning algorithm is used to build statistical tables from the corpora. This process is called the learning or training. The statistical tables consist of statistical information such as the

characteristics of well-formed sentences and the correlation between the languages. During translation, the collected statistical information is used to find the best translation for the input sentences. This translation step is called the decoding process [5].

In SMT, the core process (transfer) includes a translation model which takes as input SL words or word sequences (phrases) and produces TL words or word sequences as an output. The following stage includes a language model which synthesizes the sets of TL words in meaningful strings which are meant to be equivalent to the input sentences. The preceding (analysis) phase is represented by the conventional process of matching individual words or word sequences of input SL text against entries in the translation model [1].

2.3.2 Example-Based Machine Translation (EBMT)

EBMT is a translation method that retrieves similar examples (pairs of source phrases, sentences, or texts and their translations) from a database of examples adapting the examples to translate new input [2]. EBMT is the main subject of this research and it will be explained in details in the next section.

3. Example-Based Machine Translation

EBMT system rests on the idea that similar sentences will have similar translations. It uses past translation examples to generate a translation for a given SL text. The system maintains an example-base (EB) consisting of translation examples. When a SL sentence is given to the system, the system retrieves a similar SL sentence from the EB with its translation. Then it adapts the example to generate the TL sentence for the input sentence.

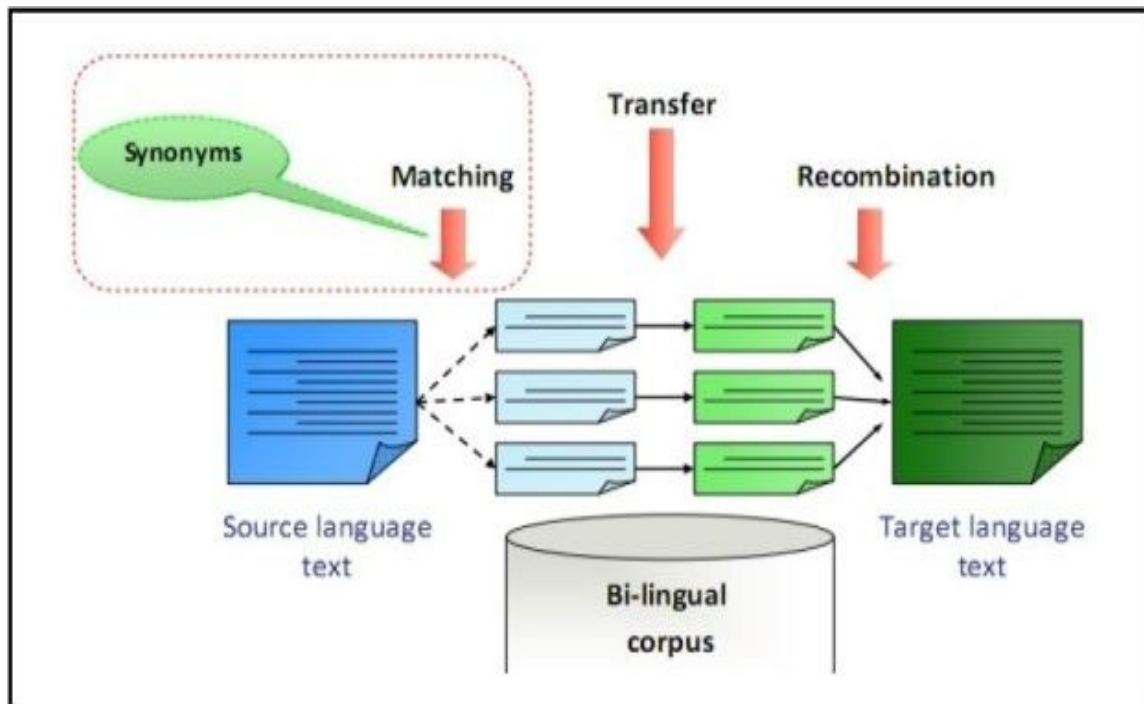


Figure 1 EBMT Working Strategy

The system has two main modules 1) retrieval and 2) adaption [4].

There are three tasks in EBMT:

- Matching fragments against existing examples.
- Transferring (Identifying the corresponding translation fragments).
- Recombining the fragments to give the target text [2].

3.1 Stages of EBMT

In general, there are four stages of work in EBMT. There are example acquisition, example base management, example application, and target sentence synthesis. Example acquisition is about how to obtain examples from parallel bilingual corpus. The example base management is about how examples are stored and maintained.

The example application stage is about how examples are used to facilitate translation, which involves the decomposition of an input sentence into examples and the transformation of source texts into target texts in terms of existing translation. The sentence synthesis is to generate a target sentence by putting the converted examples into a smoothly readable order, aiming at improving the readability of the target sentence after conversion [2].

3.2 Advantages of EBMT

There are several main advantages from using EBMT:

• Improvement

EBMT has no rules, thus improvement is effected simply by adding appropriate examples to the database. In other words, EBMT is easily upgraded.

• Translation speed

EBMT directly returns a translation by adapting the examples without reasoning through a long chain of rules. In EBMT, deep semantic analysis is avoided because it is assumed that translations that are appropriate for a given domain can be obtained using domain-specific examples.

• Translation Accuracy

In EBMT, a reliability factor is assigned to the translation result according to the distance between the input and the similar examples found. In other words, EBMT can tell when its translation is inappropriate [6].

3.2 Drawbacks of EBMT

Although the quality of translation improved as more examples were added to the database, but there is a limit after which further examples do not improve the quality. There may be cases where performance starts to decrease and retrieval from the example database will be slow. The reason is because of storing and accessing of a large corpus of examples, and of matching an input phrase or sentence against this corpus [7].

Thus in the proposed method, C B⁺ tree will be used in order to avoid this problem and to design a special dictionary for the source language sentences that works on:

- Provide efficient time for getting the translation of the source language sentence.
- Provide efficient memory usage in storing the source language sentences.

4. B⁺ Tree

B⁺ tree is a data structure consists of nodes that linked by pointers (internal nodes), a special node called the root, and

leaves. It has a unique path to each leaf, and all paths are equal in length. Each node of the tree contains an ordered list of reference values and pointers to lower level nodes in the tree. These pointers can be thought of as being between each of the references values. It stores keys only at leaves, and stores reference values in other internal nodes. The key search is guided via the reference values, from the root to the leaves.

To search for or insert an element into the tree, the root of the B⁺ Tree should be the starting point because it represents the whole range of values in the tree, where every internal node is a subinterval. We are looking for a value k in the B⁺ Tree. Starting from the root, the leaf which may contain the value k is looked for. At each node, the adjacent reference values are found that the searched-for value is between and follows the corresponding pointer to the next node in the tree. An internal B⁺ Tree node has children where every one of them represents a different sub-interval. Recursion eventually leads to the desired value or the conclusion that the value is not present.

B⁺ tree is often used in the implementation of database indexes, such that each record will be stored in the database. The reference number and the key of that record will be stored in the B⁺ tree. To reach a certain record, we need to know its key to get its reference number from the B⁺ tree. When we get the reference number of that record we can retrieve the required record directly and efficiently.

B⁺ tree is an arranged and balanced tree, see Figure 2. This is why it is so fast in retrieving the required data [8].

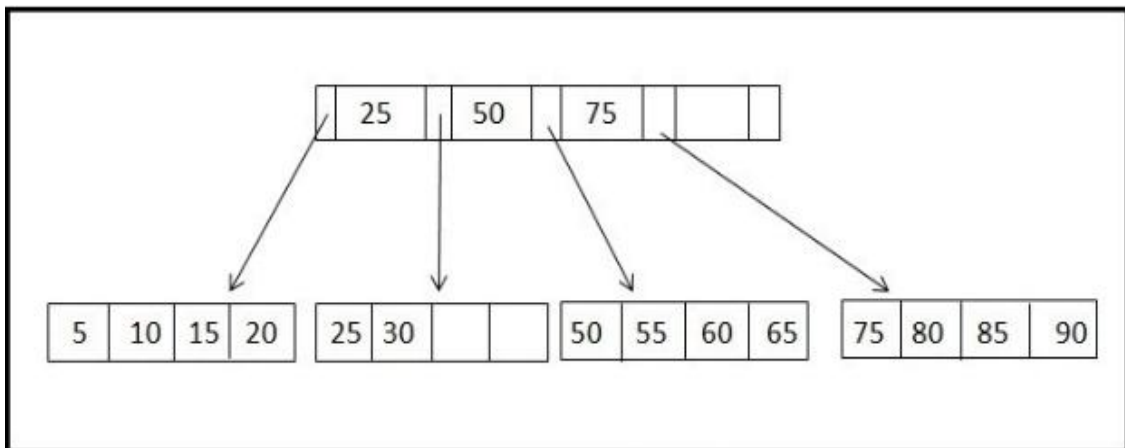


Figure 2 An example of B⁺ Tree

4.1 Insertion and deletion in B⁺ Tree

To insert a value in B⁺ Tree, the following steps should be taken:

- Find the leaf in B⁺ Tree to insert the value into.
- If the leaf is full, the node should be split and the index should be adjusted accordingly.

To delete a value from B⁺ Tree, the following steps should be taken:

- Find the leaf in B⁺ Tree to delete the value from.
- Delete the specified value.
- If the number of the remaining values in the node is less than half-full, the index should be adjusted accordingly [9].

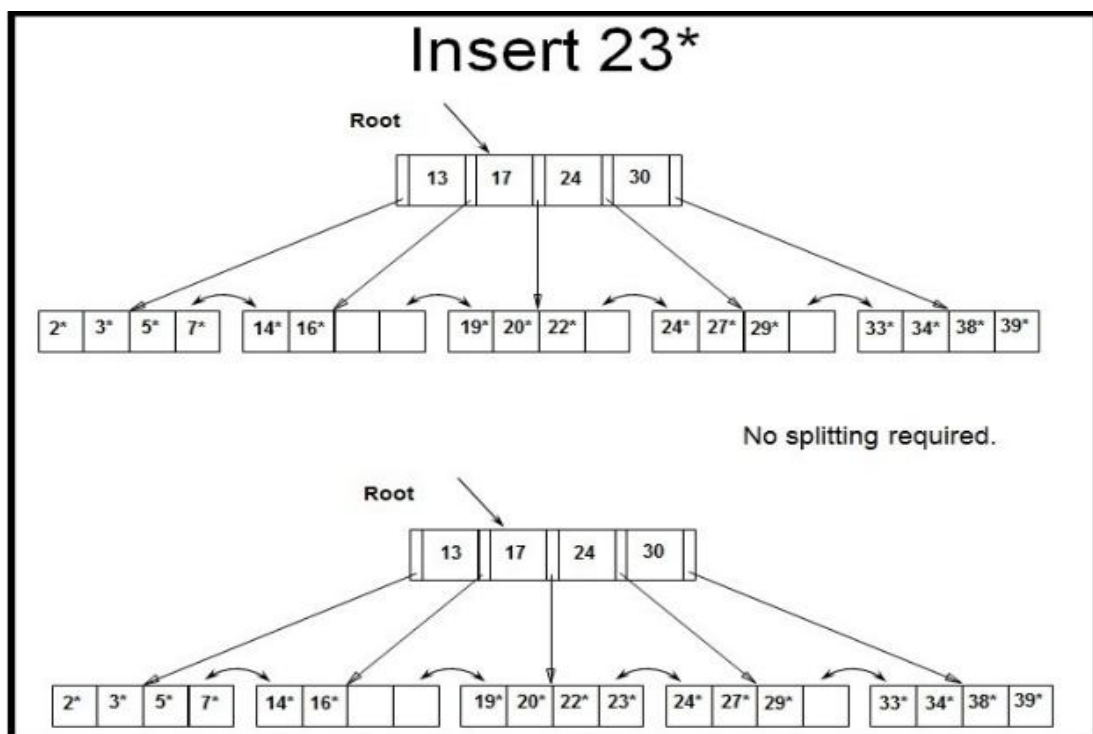
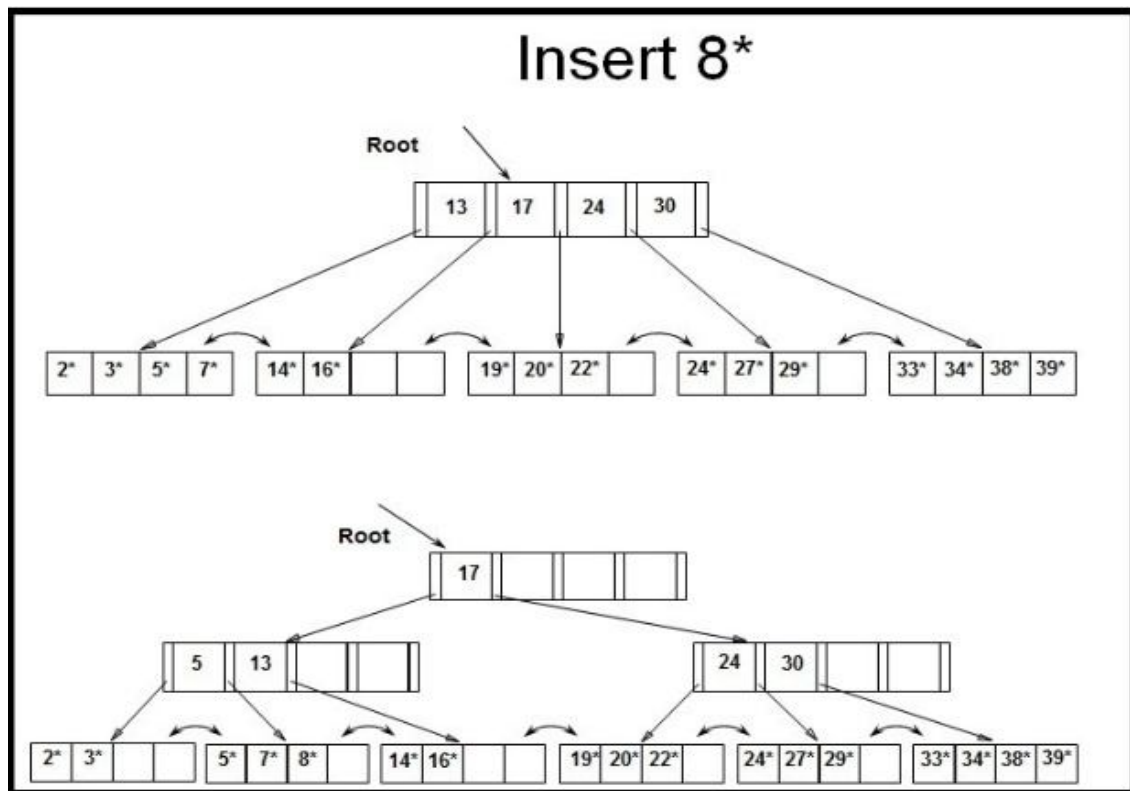
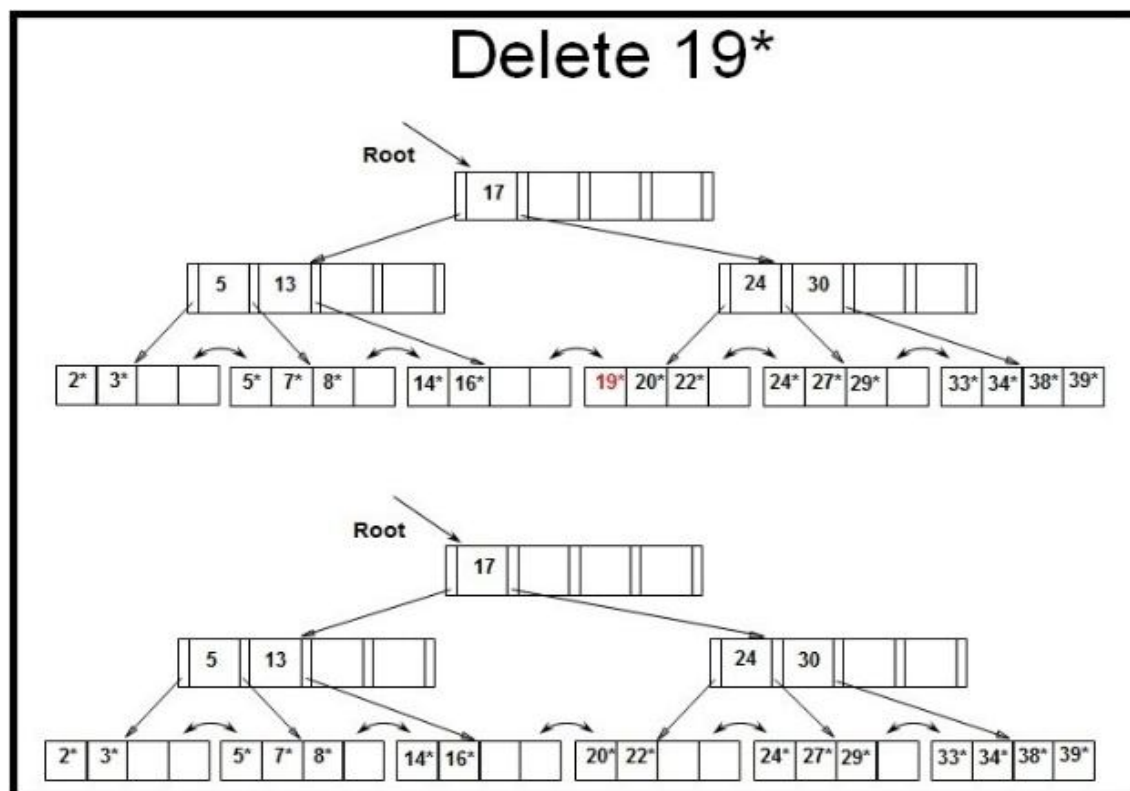
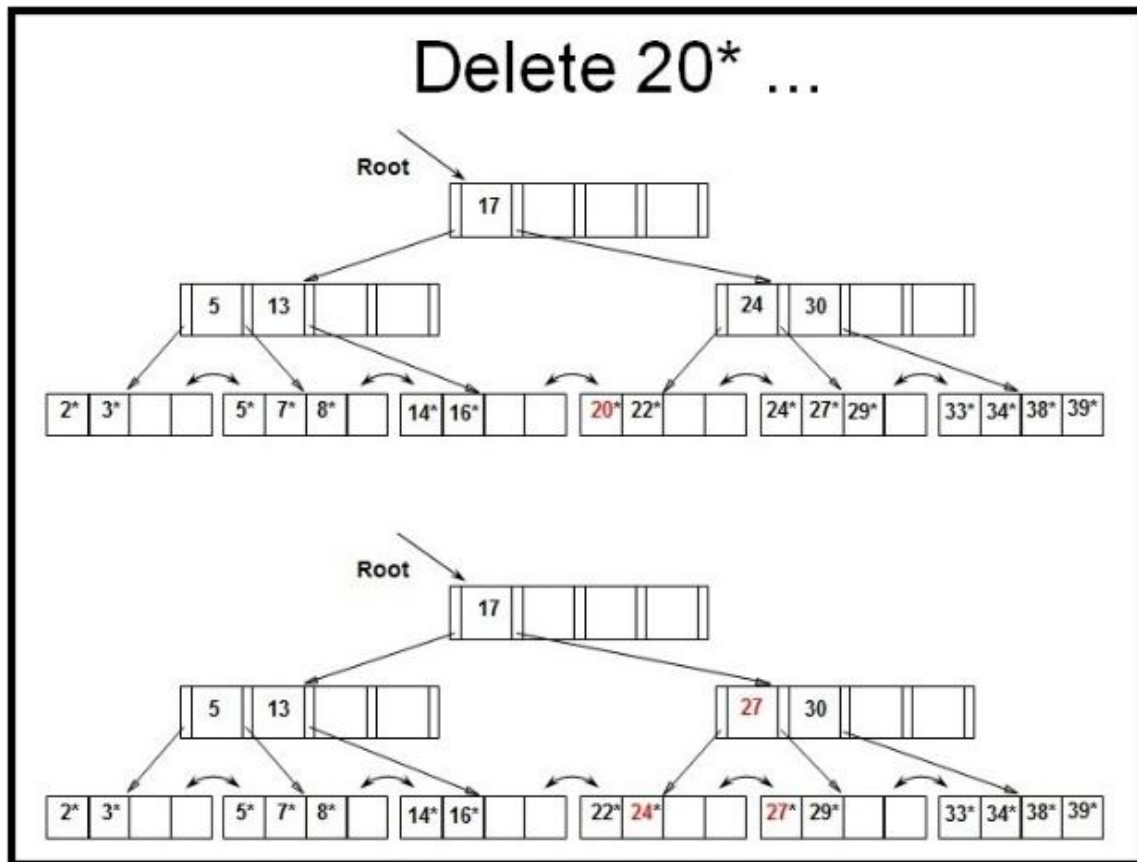


Figure (3-A) An example of insertion in B⁺ Tree

Figure (3-B) An example of insertion in B⁺ TreeFigure (3-C) An example of deletion in B⁺ Tree

Figure (3-d) An example of deletion in B⁺ Tree

5. Description of the proposed method

In this research, a new approach is suggested for designing EBMT system by using B^+ tree. The proposed system depends mainly on the examples stored in the Example Base (EB) to get the translation of the input sentence. It will search for the input sentence in the (EB). If the input sentence is found in the (EB), then the system will retrieve its corresponding translation. If the input sentence is not found among the examples in the (EB), it will be partitioned into sub-sentences and compared against the examples in the (EB). If these sub-sentences are found in the (EB), the system will retrieve its corresponding translations. If these sub-sentences are not found in the (EB), the EBMT system will depend on word by word analysis of the input sentence to get the translation.

Figure (4) shows the architecture of the proposed method.

The user interface is responsible on interaction between the proposed system and the user in ease form (since a visual programming language is used). The user can update the contents of the lexicon through user interface by removing or adding a new English word with its information (like: type, specific type, number, sex, suffix, prefix,..., etc.). Also the user can update the contents of the EB through user interface by removing, updating, or adding a new English example with its Arabic translation.

The input to the proposed system will be an English text consists of sentences (a sentence is considered to be a set of words separated by a stop mark ".", "?", or "!").

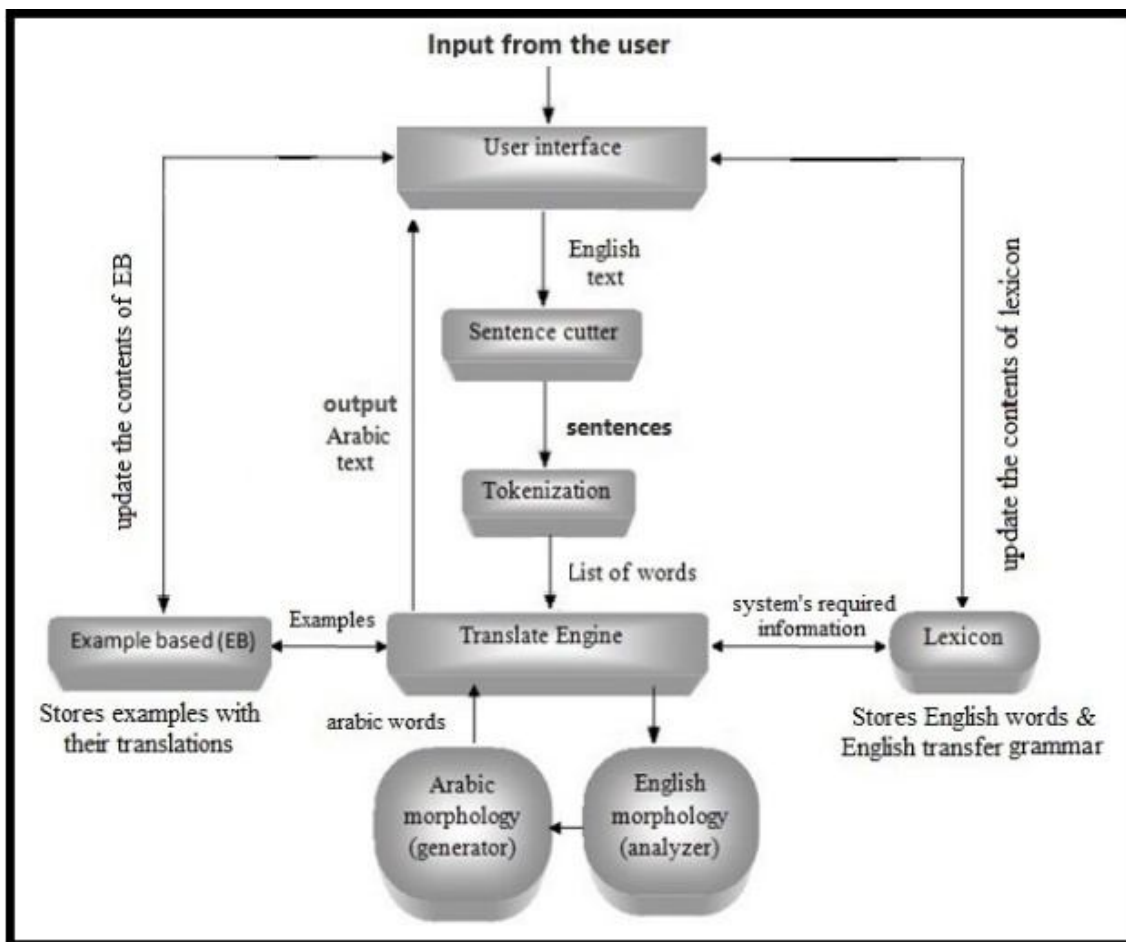


Figure 4 The architecture of the proposed method

The sentence cutter is responsible on producing these sentences. Tokenization part of the proposed system is used for converting the sentence to a list of words. The other parts of the proposed system will be discussed with more details in the following sections.

5.1 Lexicon

Lexicon is an important part in any linguistic system. It is responsible on providing the system with its required information. The lexicon of the proposed method is represented by using two databases (with their index trees). One database (DB1) is used for storing the English words with its information such that the key for BT1 that is the English stem. The other database (DB2) is used for storing the English transfer grammars such that the key for BT2 that consists of three parts, the first and third parts are digits that correspond to the types of words or sub-sentences while the middle part is string.

5.2 Example Base (EB)

The EB is an important component in

an EBMT system. It handles the storage to support the translation process (fully automatic or human-aided). Thus, an efficient EB must be capable of handling a massive volume of examples at an adequately high speed. EB is used for storing the English examples with their Arabic translations for a particular domain (in our work we choose the computer science field). EB is represented by using one database (DB3) such that the first keyword of the input example will be the key in its index tree (BT3).

The Examples are stored in the EB in a manner that prevent redundancy to provide efficient memory usage and to minimize search time. In general, if one example consists of [word₁, word₂, word₃] with its translation T1 and another example consists of [word₁, word₂] with its translation T2, then there is no need to restore the second example. Only T2 need to be added to the DB3, see Figure 5-a. If another example consists of [word₁, word₂, word₄, word₅] with its translation T3 then only word₄ and word₅ will be added to the DB3 with T3, see Figure 5-b.

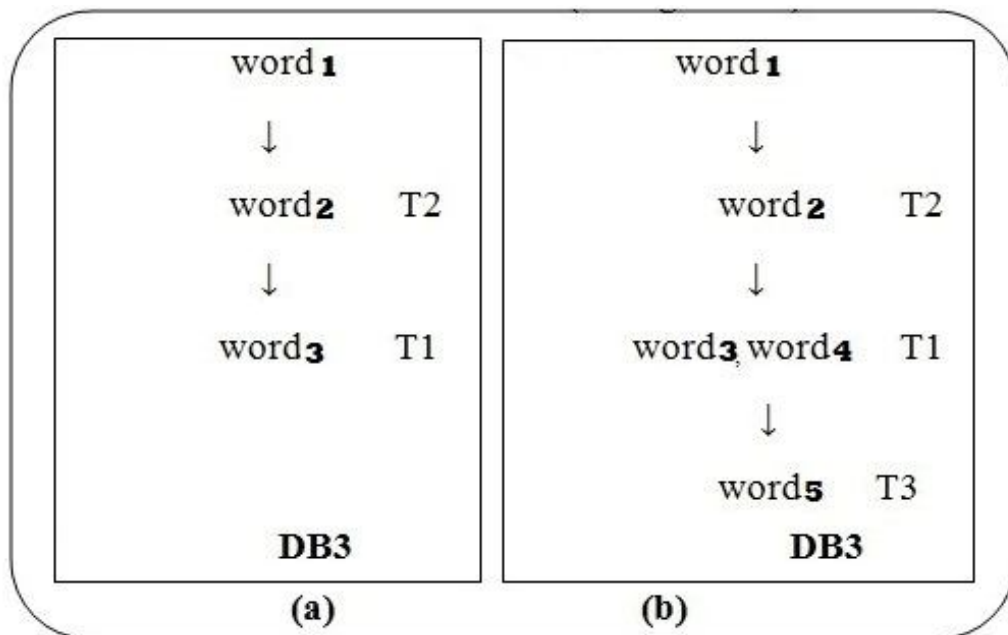


Figure 5 Preventing redundancy in the EB of the proposed method

The Examples are stored in the EB in a manner that prevent redundancy to provide efficient memory usage and to minimize search time. In general, if one example consists of [word₁, word₂, word₃] with its translation T1 and another example consists of [word₁, word₂] with its translation T2, then there is no need to restore the second example. Only T2 need to be added to the DB3, see Figure 5-a. If another example consists of [word₁, word₂, word₄, word₅] with its translation T3 then only word₄ and word₅ will be added to the DB3 with T3, see Figure 5-b. of corresponding word or sub sentence, and K is the keyword of the input sentence. Let's take a simple example: "Ahmed went to the school". The key will be "1 went to 2" such that 1 means single male proper noun, and (went to) is the keyword of the input sentence, and 2 refers to single determiner noun. This key will correspond to the transfer grammar

5.3 Morphology

English morphology is responsible on extract the stem for English word by removing its suffix or prefix and removing the changes that occur during adding these affixes according to the spelling rules of English language. Arabic morphology is used to generate Arabic words according to the analyzing of English morphology.

5.4 Translate Engine

Translate engine is responsible on converting the source English sentence into the target Arabic sentence by using the information supported by the lexicon and examples supported by EB. The translate engine search on the EB. If the input sentence is not found among the examples of the EB, then it extracts a key from the input sentence. The key is composed of the form XKY. Where X and Y may be digits that refer to the type

6. Algorithms of the proposed method

In this section a focus is only on the algorithms of the Example Base (EB) component of the proposed system, which describe how B⁺ tree method will be used to store the examples in the EB and prevent redundancy, see algorithm 1.

Algorithm1: "store_english_sen"

Input: S: English sentence.

Process:

Begin

1. If the file "DB" exist then open DB and its BT otherwise create them.
2. Convert S to a list of words (List1).
3. Get the first keyword found in List1 to be the Key.
4. Search in (BT) for the Key.
5. If (not found) then
 - 5.1. Call "add_new_sentence" function to get Ref. /*see algorithm 2*/
 - 5.2. Insert the Key in (BT) with the reference (Ref).

Else

- 5.3. Call "check_found_sentence" function. /*see algorithm 3*/

End.

Algorithm2: "add_new_sentence"

Input: list of words (List1), Arabic translation (T1).

Output: database reference number (Ref).

Process:

Begin

1. Compute the length of List1 to be N.
 2. Remove the first word (W) from List1.
 3. If (List1==[]) then
 - 3.1. Insert the term: word ([p (W, null, N, T1)]) to DB at Ref.
 - 3.2. Return (Ref).
 - Else
 - 3.3. Insert the term word ([p1 (W, NRef)]) to DB at Ref. /* NRef is the reference of next word */
 - 3.4. Goto 2.
- End.

Algorithm3: "check_found_sentence"

Input: list of English words (List1), Arabic translation (T1).

Process:

Begin

1. If all the words of List1 are found in DB and the sentence has a translation then
 - 1.1. Ask the user if he want to replace it.
 - 1.2. If answer=yes then replace the term of the last word of List1 with the new translation T1.
2. If all the words of List1 are found in DB but with no translation then
 - 2.1. Store the translation T1 with the last word of List1 found in DB.
3. If all the words of List1 are found in DB except the last word then
 - 3.1. Store the last word in DB with the translation T1.
4. If some beginning words of List1 are found in DB then
 - 4.1. Store the remaining words of List1 in DB (except the last word) with the reference of their next word.

- 4.2. Store the last word of List1 in DB with the translation T1.

End.

7. System Implementation, Test and Results

In this section, some examples that describe only the Example Base (EB) part of the proposed system are taken. Then the results of an experiment that test the accuracy of the proposed system will be shown.

Example1:

If we want to get the translation for the source language sentence from the (DB), and suppose it is a new sentence such as: **"He will be able to play soccer"**, then we put the first Keyword of the sentence (able) as a key in B⁺ tree (BT). We compute the length (7) of the sentence and give it a new translation. The sentence will be stored with its translation in DB, as shown in Figure 6.

<u>BT</u>	<u>References</u>	<u>DB</u>
	0001	Word([P("He", 0010, 0, " ")])
	0010	Word([P("will", 0011, 0, " ")])
	0011	Word([P("be", 0100, 0, " ")])
"able"→	0100	Word([P("able", 0101, 0, " ")])
	0101	Word([P("to", 0110, 0, " ")])
	0110	Word([P("play", 0111, 0, " ")])
	0111	Word([P("soccer", ~0, 7, " سيكون قادراً على لعب كرة القدم")])

Figure 6 Representation of the sentence "He will be able to play soccer"

Example2:

If we want to get the translation for a sentence such that all its words are already found in (DB) but with no translation, such as: **"He will be able to play"**. We give it a

new translation and store it in (DB). It will be as shown in Figure 7.

<u>BT</u>	<u>References</u>	<u>DB</u>
	0001	Word([P("He", 0010, 0, " ")])
	0010	Word([P("will", 0011, 0, " ")])
	0011	Word([P("be", 0100, 0, " ")])
"able"→	0100	Word([P("able", 0101, 0, " ")])
	0101	Word([P("to", 0110, 0, " ")])
	0110	Word([P("play", 0111, 6, "سيكون قادراً على اللعب")])
	0111	Word([P("soccer", ~0, 7, "سيكون قادراً على لعب كرة القدم"), P("tennis", ~0, 7, "سيكون قادراً على لعب التنس")])

Figure 7 Representation of the sentence "He will be able to play"

Example3:

If we want to get the translation for a sentence such that all its words are already found in (DB) except the last word, such as: **"He will be able to play**

tennis". We will store the last word and the new translation in (DB), and store at the previous word the new reference. It will be as shown in Figure 8.

<u>BT</u>	<u>References</u>	<u>DB</u>
	0001	Word([P("He", 0010, 0, " ")])
	0010	Word([P("will", 0011, 0, " ")])
"able"→	0011	Word([P("be", 0100, 0, " ")])
	0100	Word([P("able", 0101, 0, " ")])
	0101	Word([P("to", 0110, 0, " ")])
	0110	Word([P("play", 0111, 6, "سيكون قادراً على اللعب")])
	0111	Word([P("soccer", ~0, 7, "سيكون قادراً على لعب كرة القدم"), P("tennis", ~0, 7, "سيكون قادراً على لعب التنس")])

Figure 8 Representation of the sentence "He will be able to play tennis"

Example 4:

If we want to get the translation for a sentence such that some of its words are already found in (DB), such as: **“He will be able to play soccer tomorrow morning”**. We will store the remaining words and the new translation in (DB), and store the references of the next words. It will be as shown in Figure 9.

Example5:

If we want to get the translation for a sentence such that some of its middle words are found in (DB), such as: **“Definitely he will be able to play”**. Only the not found words will be stored in (DB) and give the sentence a new translation. It will be as shown in Figure 10.

BT	References	DB
	0001	Word([P("He", 0010, 0, " ")])
	0010	Word([P("will", 0011, 0, " ")])
	0011	Word([P("be", 0100, 0, " ")])
"able"→	0100	Word([P("able", 0101, 0, " ")])
	0101	Word([P("to", 0110, 0, " ")])
	0110	Word([P("play", 0111, 6, "سيكون قادراً على اللعب")])
	0111	Word([P("soccer", 1000, 7, "سيكون قادراً على لعب كرة القدم"), P("tennis", ~0, 7, "سيكون قادراً على لعب التنس")])
	1000	Word([P("tomorrow", 1001, 0, " ")])
	1001	Word([P("morning", ~0, 9, "سيكون قادراً على لعب كرة القدم غداً صباحاً")])

Figure 9 Representation of the sentence “He will be able to playsoccer tomorrow morning”

BT	References	DB
	0001	Word([P("He", 0010, 0, " ")])
	0010	Word([P("will", 0011, 0, " ")])
"able"→	0011	Word([P("be", 0100, 0, " ")])
	0100	Word([P("able", 0101, 0, " ")])
	0101	Word([P("to", 0110, 0, " ")])
	0110	Word([P("play", 0111, 6, "سيكون قادراً على اللعب"), P("play", ~0, 7, "بالتأكيد سيكون قادراً على اللعب")])
	0111	Word([P("soccer", 1000, 7, "سيكون قادراً على لعب كرة القدم"), P("tennis", ~0, 7, "سيكون قادراً على لعب التنس")])
	1000	Word([P("tomorrow", 1001, 0, " ")])
	1001	Word([P("morning", ~0, 9, "سيكون قادراً على لعب كرة القدم غداً صباحاً")])
	1010	Word([P("Definitely", 0001, 0, " ")])

Figure 10 Representation of the sentence “Definitely he will be able to

An experiment is made to test the accuracy of the proposed system by comparing the translations of 30 English sentences translated by using the proposed system against the translations of the same 30 sentences translated by an Instructor as in [10].

The result of the comparison is shown in Table 1. In this table, it is clear that out of 30 submitted English sentences, there are 24 sentences whose translation by using the proposed system is identical to the Instructor's translations.

Table 1: Comparison of results between Instructor's translations and the proposed system translations

No.	English sentences	Instructor's translation	Translation of the proposed system
1	They are students	هم طلاب	هم طلاب
2	Sally is invited to the party	سالي مدعوة الى الحفلة	سالي مدعوة الى الحفلة
3	I am drinking water	انا اشرب ماء	انا اشرب الماء
4	Mary was quite happy	كانت ماري سعيدة جدا	كانت ماري سعيدة جدا
5	Was tom going home?	هل كان توم ذاهبا الى البيت؟	هل كان توم ذاهبا الى البيت؟
6	Were they happy?	هل كانوا سعداء؟	هل كانوا سعداء؟
7	Layla does not like tea	لا تحب ليلي الشاي	ليلى لا تحب الشاي
8	Do they play football?	هل يلعبون كرة القدم؟	هل يلعبون كرة القدم؟
9	He did his work well	قام بعمله بشكل جيد	قام بعمله بشكل جيد
10	Is that your book?	هل ذلك كتابك؟	هل ذلك كتابك؟
11	Can you speak English?	هل تستطيع التحدث باللغة الانكليزية؟	هل تستطيع التحدث باللغة الانكليزية؟
12	Do you agree?	هل توافق؟	هل توافق؟
13	We have arrived early today	وصلنا مبكرين هذا اليوم	وصلنا مبكرين هذا اليوم
14	I want to have my hair cut	اريد ان اخلق شعري	اريد ان اخلق شعري
15	I need some more paper	احاج الى ورق اكثر	احاج الى ورق اكثر
16	Will you have a cup of tea?	هل ترغب بفنجان شاي؟	هل ترغب بفنجان شاي؟
17	May I help you?	هل يمكن لي ان اساعدك؟	هل اساعدك؟
18	You should try that new restaurant	عليك ان تجرب ذلك المطعم الجديد	يجب ان تجرب ذلك المطعم الجديد
19	You shouldn't drink and drive	عليك ان لا تشرب وتقوم السيارة	يجب ان لا تشرب وتقوم السيارة
20	The students should have gone	كان على الطلاب ان يذهبوا	كان على الطلاب ان يذهبوا
21	He said they would go home	قال بانهم سيذهبون الى البيت	قال بانهم سيذهبون الى البيت
22	Would you mind opening the window?	هل تمنع من فتح الشباك؟	هل تمنع من فتح الشباك؟
23	Would you rather have tea or coffee?	هل تفضل الشاي ام القهوة؟	هل تفضل الشاي ام القهوة؟
24	You can go as soon you like	تستطيع الذهاب متى تشاء	تستطيع الذهاب متى تشاء
25	I want to get two cans of sardines	اريد ان احصل على علبتين من السردين	اريد ان احصل على علبتين من السردين
26	I can smell something burning	اشم رائحة شئ يحترق	استطيع ان اشم رائحة شئ يحترق
27	He can be nervous	قد يكون عصبي المزاج	قد يكون عصبي المزاج
28	Selma cannot be hungry	لا يمكن لسلمي ان تكون جائعة	لا يمكن لسلمي ان تكون جائعة
29	She couldn't finish her book last night	لم تستطع انتهاء كتابها ليلة البارحة	لم تستطع انتهاء كتابها ليلة البارحة
30	Will you play football?	هل تلعب كرة القدم؟	هل تلعب كرة القدم؟

Table 2 shows the precision of the proposed method that resulted from the experiment:

$$\text{Precision} = \frac{\text{no. of sentences correctly translated by EBMT}}{\text{the total no. of sentences translated by EBMT}}$$

$$\text{Precision} = \frac{24}{30} = 0.8 \text{ or } 80\%$$

Table 2 shows that the precision scored by the proposed system is about 80%. There were few sentences that had different translations from the Instructor's translations. The experiment shows a great convergence between Instructor's result and EBMT result.

Table 2: Precision of the proposed method

No. of Sentences tested	Correct Translation	Unexpected Translation	Precision in %
30	24	6	80%

8. Discussion

In this section, comparison between the proposed EBMT system and the traditional EBMT is made:

8.1 Translation speed

The proposed EBMT system is faster than the traditional EBMT. The reason is because the examples are stored in the EB in a way that reduces its redundancy to provide efficient memory usage and to speed up the search by using B⁺ Tree.

8.2 Translation accuracy

The proposed EBMT has more accuracy than the traditional EBMT. The reason is because in the proposed method, the translation of each sentence is always stored with the last word of each one in DB. In traditional EBMT, the sub-sentences are stored with its translations in DB such that the translation of a sentence is generated by combining the translations of its composing sub-sentences, so the translation resulted could be weak and of low quality.

Precision is the number of sentences accurately translated by EBMT divided

by the aggregate number of sentences translated by EBMT.

9. Conclusions

In this paper the following points can be concluded:

- Using the Example Base (EB) and the examples stored in it will increase the translation speed and accuracy.
- Prevent the redundancy of the examples in the EB or even the sub examples will provide efficient memory usage.
- Using B⁺ tree for representing the EB for examples that may found in a particular field will provide an efficient search time.
- Using a lexicon that based on stems of English words and depends on morphology will provide efficient memory usage.
- Using a lexicon for English words and an English transfer grammar will reduce the number of examples that need to be stored in EB and make the system more flexible.

References

- [1] Li Peng, "A Survey of Machine Translation Methods", 2013, <http://www.iaesjournal.com/online/index.php/TELKOMNIKA/article/viewFile/2780/>
- [2] Vani K, "Example Based Machine Translation", 2010, <http://dSPACE.cusat.ac.in/jspui/bitstream/123456789/3623/1/EBMTorginal>
- [3] Harjinder Kaur, Dr. Vijay Laxmi, "A SURVEY OF MACHINE TRANSLATION APPROACHES", 2013, <http://ijsetr.org/wp-content/uploads/2013/07/IJSETR-VOL-2-ISSUE-3-716-719>
- [4] Jaganadh G, "Man to Machine A tutorial on the art of Machine Translation", 2010, <http://www.slideshare.net/jaganadhg/a-tutorial-on-machine-translation>
- [5] Antony P. J., "Machine Translation Approaches and Survey for Indian Languages", 2013, <http://www.aclclp.org.tw/clclp/v18n1/v18n1a3>
- [6] Andrea Schuch, "EBMT Based upon Two-Dimensional Alignment", 2010, <http://iaaa.nl/rs/PDF/SchuchThesis>
- [7] Khan Md. Anwarus Salam, Setsuo Yamada, Tetsuro Nishino, "Example-Based Machine Translation for Low-Resource Language Using Chunk-String Templates", 2011, <http://www.mt-archive.info/MTS-2011-Anwarus-Salam>
- [8] Goetz, Graefe, "B-tree indexes, interpolation search, and skew", Chicago Illinois, USA, 2006.
- [9] Mike Franklin, "B+ Trees", 2006, <https://www.cs.umd.edu/class/spring2006/cmsc424>
- [10] Abdul- Hassan Sh. Qassim, "Translation Grammatically viewed", English department, University of Baghdad.