English-Thai Example-Based Machine Translation using *n*-gram model

Nattapol KRITSUTHIKUL, Arit THAMMANO, and Thepchai SUPNITHI

Abstract — **The necessity on exchanging information among countries become a major task in information based society. Machine translation is an application that enables users to communicate each other without language barrier problem. With the great support on computer's efficiency, corpus-based technology becomes a fundamental concept for developing software based on a large amount of data. We introduce the first example-based English to Thai machine translation using *n*-gram model and implemented the system. Some advantages and disadvantages of this method are discussed.**

I. Introduction

THERE are several approaches on Machine Translation (MT) research. The most of distinctive methods are rule-based [1] and corpus-based methods. Research on the corpus-based approach has emphasized on the important of text corpora as fundamental sources of data for linguistic and knowledge database. There have been two major approaches in the corpus-based MT: statistical-based approach [2] and example-based approach [3]. Currently, a lot of English to Thai machine translation software products, such as ParSit [4], AgentDict [5], were launched to end users. All of them use the rule-based approach, whose all knowledge from linguists is externalized as a set of inference rules. This approach has several drawbacks related to time consumption and rule conflict. Then, corpus-based MT becomes much more interesting topics in NLP research field. However, there are very few researches on corpus-based MT in Thai. Some research groups try to apply corpus as a memory for editing the incorrect answer from rule based results [6]. In this paper, we concentrate on corpus based MT in example-based approach. We generate the possible patterns by applying *n*-gram model.

This paper is organized as follows. Section 2 focuses on our system architecture, all components in the system, and techniques in analyzing and generating sentences. Section 3 mainly explains the corpus used in this system. Section 4 illustrates the implementation design and tool. Finally, section 5 shows a conclusion and future work.

Nattapol KRITSUTHIKUL and Arit THAMMANO, are with the Computational Intelligence Laboratory, Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Road, Bangkok 10520, Thailand (e-mail: s4467499@kmitl.ac.th and arit@it.kmitl.ac.th)

Nattapol KRITSUTHIKUL and Thepchai SUPNITHI, is with the Research and Development in Information Division, National Electronic and Computer Technology Center, Thailand (e-mail: thepchai@nectec.or.th)

II. System Architecture

The system architecture is shown in Figure 1. Our system composes of two main components: analysis component and generation component. We apply the *n*-gram approach to achieve varieties of patterns based on partial sentences. When a system receives a source sentence, it will be passed to match with data in our prepared bilingual parallel corpus. All appropriate matching alternatives (sentences or partial sentences) will be retrieved and sent to find the most appropriate target sentences. Monolingual corpus will help analyzing and generating an appropriate alternatives in each component. Bilingual corpus composes of parallel sentences and parallel partial sentences. All data in this corpus functions as transfer rules. In the current version, we collect partial sentences manually from linguists.

A. The n-gram approach

The *n*-gram language model is based on the following assumption: the n^{th} word is only related to its preceding *n*-1 words. Therefore the probability estimation of the language model $P(w)$ can be written as $P(w_n|w_1,...,w_{n-1})$. For a sentence with n words, given the candidates $w_1,w_2,...,w_n$ from the bilingual corpus, the probability of the whole sentence is calculated by the equation (1).

$$P(w) = \sum_{i=1}^{N} P(w_i \,|\, w_{i-n+1},...,w_{i-1}) \qquad (1)$$

For a large amount of text corpora, the probability of $P(w_n|w_1,...,w_{n-1})$ can be estimated from the maximum likelihood principle by the equation (2):

$$P(w_n \,|\, w_1,...,w_{n-1}) = \frac{C(w_1,...,w_n)}{C(w_1,...,w_{n-1})} \qquad (2)$$

where $C(w_1,...,w_{n-1})$ and $C(w_1,...,w_n)$ represent the occurrence number of the word string $w_1,...,w_{n-1}$ and $w_1,...,w_n$, respectively.

B. N-gram analysis component

The *n*-gram analysis component is aimed to analyze a source sentence by breaking it into all possible pieces of sentences and to select the most suitable alternative. The main function in the analysis component is matching algorithm. Matching

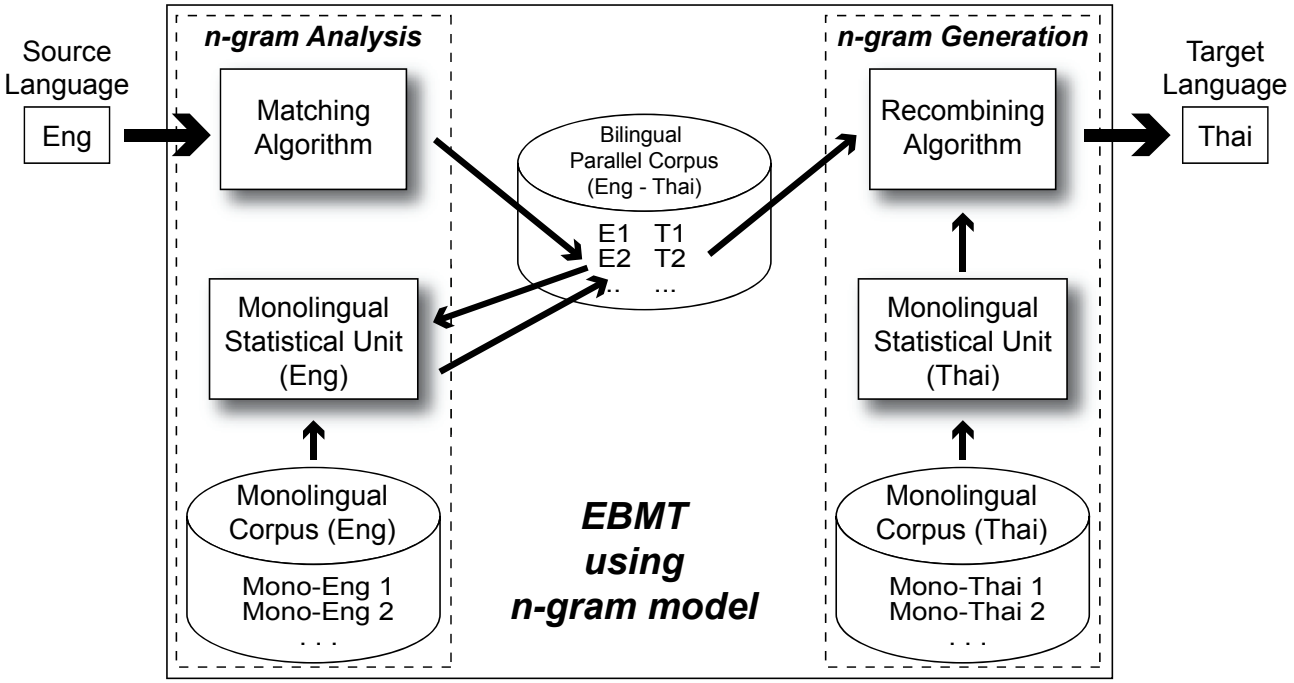

Fig.1. System Overview of EBMT using n-gram model

algorithm is applied to break a sentence into a combination of sub-sentences that composes of the longest sub-sentence. The longest sub-sentence is defined by matching with all data in bilingual corpus. It will lead us to find the least number of fragmentations in the sentence.

The matching algorithm is defined as follows.

Step 1: Define *Fragment_Set = {SL}* and *Result_Set = { }*
Step 2: Generate sub-fragment Sf_i from *Fragment_Set* by segmenting groups of words that $next(w_i) \neq w_{i+1}$
Step 3: For each Sf_i that has more than one elements, find the maximum sub-sentence Max_Sf_i in Sf_i
Step 3.1: Push Max_Sf_i into *Result_Set*
Step 3.2: Delete Max_Sf_i from *Fragment_Set*
Step 4: Repeat Step 1 until each sub-fragment Sf_i has only one element
Step 5: Return Sf_i
Step 6: Return *Result_Set*

From the example sentence "*Arsenal picked up a big victory in Champions League*", we assume that there are *{a big victory}, {picked up}* and *{Champions League}* in our bilingual corpus. The matching algorithm will be processed as follows.

Step 1: *Fragment_Set = { {Arsenal picked up a big victory in Champions League} } , Result_Set = { }*
Step 2: Since $next(w_i) = w_{i+1}$ for all w_i , sub-fragment Sf_i has only one sub-fragment that is Sf_1 *= {Arsenal picked up a big victory in Champions League}*
Step 3: maximum sub-sentence Max_sf_1 *= {a big victory}*
Step 3.1: *Result_Set ={ {a big victory} }*
Step 3.2: *Fragment_Set = { {Arsenal picked up}, {in Champions League} }*
Step 4: *Fragment_Set = { {Arsenal picked up}, {in Champions League} } , Result_Set = { {a big victory} }*
Step 5: Since $next(w_i) \neq w_{i+1}$ at w_i= up, there are two sub-fragments, Sf_1 *= {Arsenal picked up}* and Sf_2 *= {in Champions League}*
Step 6: maximum sub-sentence Max_sf_1 *= {picked up}*
Step 6.1: *Result_Set ={ {picked up}, {a big victory} }*
Step 6.2: *Fragment_Set = { {Arsenal}, {in Champions League} }*
Step 7: maximum sub-sentence Max_sf_2 *= {in Champions League}*
Step 7.1: *Result_Set = { {Champions League}, {picked up}, {a big victory} }*
Step 7.2: *Fragment_Set = { {Arsenal}, {in} }*
Step 8: Since $next(w_i) \neq w_{i+1}$ at w_i= Arsenal, there are two sub-fragments Sf_1 *= { Arsenal }* and Sf_2 *={ in }*
Step 9: Return *Result_Set = { {Arsenal}, {in}, {Champions League}, {picked up}, {a big victory} }*

Results from the *n*-gram analysis will be translated to Thai based on two criteria. If an element in *Result_Set* is not a singleton, translated results in bilingual corpus will be retrieved. Otherwise, the translated results will be retrieved from dictionary. Translated results from *Result_Set* will be sent to the *n*-gram generation. From the example, *Result_Set = { {Arsenal}, {in}, {Champions League}, {picked up}, {a big victory} }* will be translated to

{ {เจ้าปืนใหญ่อาร์เซนอล} ,{ใน}, {แชมเปี้ยนลีค} ,{ได้}, {ชัยชนะครั้งใหญ่} }

C. N-gram generation component

The *n*-gram generation component is aimed to generate a target sentence by merging and ordering pieces of sentences into one sentence. The main function in the generation component is recombining algorithm. We apply Greedy Algorithm to detect the most suitable sub-sentences that should be concatenated. Recombining algorithm merges sub-sentences into a target sentence by considering the word ordering in sentence. The algorithm is explained as follows:

Step 1: Define $Fragment\_List = \{Fr_1, Fr_2, \ldots, Fr_n\}$

Step 2: [combine the maximum probability of sub-sentence and its neighbor]
For each Fr_a and Fr_b in *Fragment_List* that $1 \le a, b \le n$ and $|a-b| = 1$,
Fr_{ab} = max_combine (Fr_a , Fr_b)

Step 3: Substitute Fr_a and Fr_b with Fr_{ab} and delete Fr_a and Fr_b from *Fragment_List*
$Fragment\_List = \{Fr_1, Fr_2, \ldots, Fr_{ab}, \ldots, Fr_n\}$

Step 4: Repeat Step 1 until *Fragment_List* has only one element

Step 5: Return *Fragment_List*

For the above example, the output *Result_Set* in the *n*-gram analysis component is *{ {Arsenal}, {in}, {Champions League}, {picked up}, {a big victory} }* . We obtain the result

Fragment_List = { เจ้าปืนใหญ่อาร์เซนอล ,ใน ,แชมเปี้ยนลีค, ได้, ชัยชนะครั้งใหญ่ }

When we apply the recombining algorithm, the *Fragment_List* will be merged in step 2 by considering each sub-sentence and its neighbor (at distance 1) as shown step-by-step result in figure 2. The gray highlight identifies a pair of word/phrase that will be combined for the next step. Finally, "เจ้าปืนใหญ่อาร์เซนอลได้ชัยชนะครั้งใหญ่ในแชมเปี้ยนลีค" will be generated as translation result.

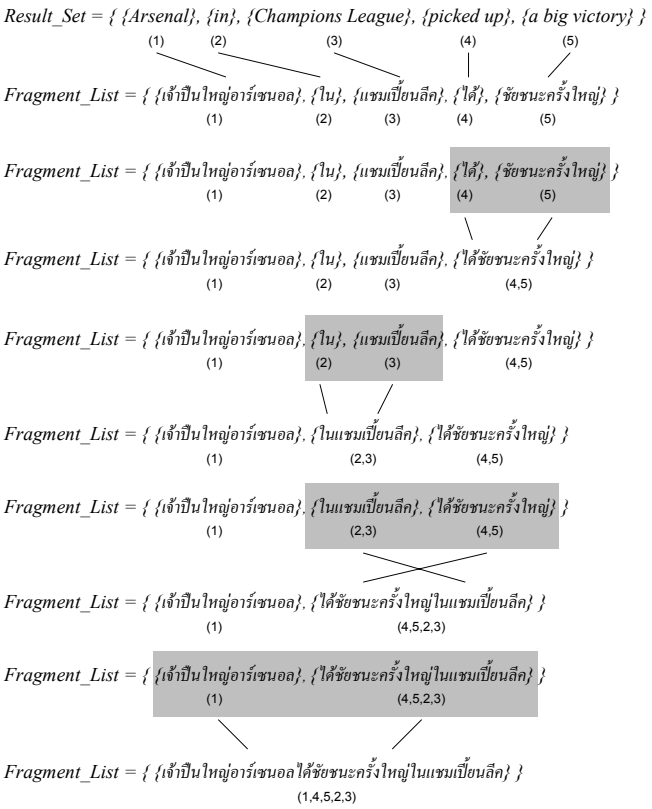

Fig. 2. *n*-gram generation component by example

III. CORPUS IN EBMT SYSTEM

In our EBMT approaches, it is necessary to prepare a large amount of parallel corpus in order to find a good matching. One of the major problems is time consumption for collecting a large amount of parallel corpus. We decide to apply monolingual corpus in each component to compensate the inadequate parallel corpus. Monolingual corpora in English and Thai for analysis phase and generation phase is much easier to collect. Since there is no word boundary in Thai, we segment sentences into words by using word segmentation tools called SWATH [7]. The occurrences of *n*-gram in each corpus are counted as statistic information and applied in the matching algorithm and the recombining algorithm. In bilingual corpus, we collect bilingual corpus based on our previous work in general domain. Sub-sentences pairs are manually constructed to increase the opportunity of matching in the analysis phase.

Currently, there are 104,893 sentences and 561,387 words in our English corpus, 16,749 sentences and 84,292 words in our Thai corpus, respectively. In parallel corpus, we collect 64,990 sentence pairs. There are 218,144 words in English and 182,243 words in Thai. The number of sentence and sub-sentence pairs is 91,068.

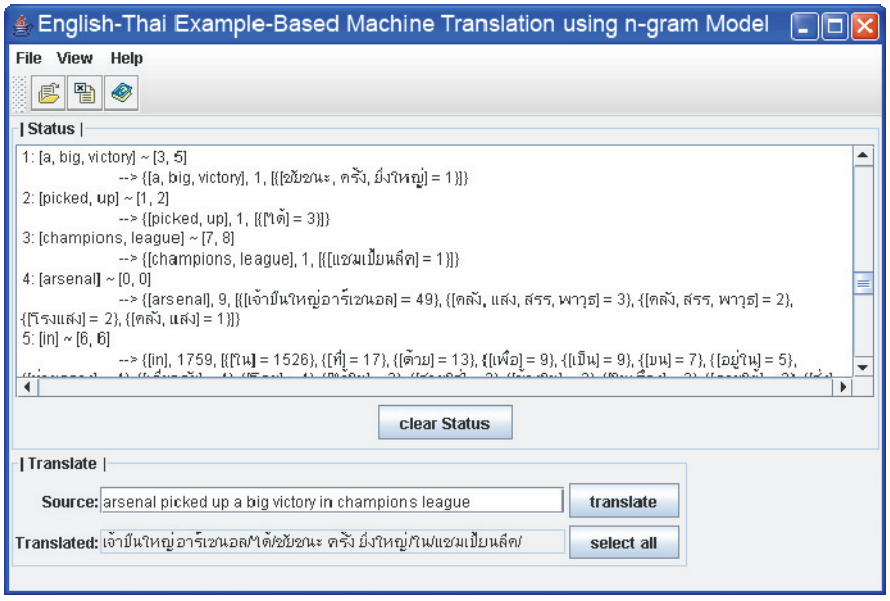

Fig. 3. EBMT using *n*-gram model application main screen

IV. Implementation

Figure 3 shows a screen of our EBMT system. When a user inputs a source sentence in the **Source** textbox and press the **translate** button. Translation process will be started. All translation processes, starting from the analysis phase to the generation phase, will be represented in the **Status** text area. The translated result will be shown in the **Translated** textbox. To increase the speed performance, all necessary information for translation, both from our corpora and our dictionary, are loaded in memory. We develop the *Trie* structure [8] at the word level to represent data in memory. This system is implemented in JAVA 5 platform (current version is jdk1.5.0_07).

V. Discussion

Example-based machine translation is a powerful technique that applies corpus-based approach. To translate from English to Thai, we investigated results and found that our methods have advantages on the complex sentence with exact partial phrase, if they were already collected in our bilingual corpus. Moreover, translation results of some metaphoric sentences and proverbs are also acceptable. However, there are some drawbacks in our method. Firstly, we need to collect partial sentence in parallel corpus as much as possible to avoid misleading results. Secondly, if suitable results cannot be found in the parallel corpus, the system will alternatively retrieve the answer from dictionary. It might lead to incorrect result because of word ambiguity and word omission. For example, a word "*the*" has no suitable translation result in Thai and will be omitted. Unfortunately, it will be translated to "*คำนำหน้านามชี้เฉพาะ*" which means "the definite article preceding proper nouns".

VI. Conclusion and Future Work

This paper explains an *n*-gram-based EBMT system for English to Thai machine translation. The translation results can be achieved by applying *n*-gram model technique in combination with information from corpus and dictionary. In the future work, three main issues will be considered. Firstly, since it is difficult to develop sub-sentences manually, we plan to analyze the possible patterns in bilingual paralleled corpus and develop an application for generating sub-sentence automatically. Secondly, If there exist more than one alternatives for each sub-sentence, we will select the most appropriate alternatives by considering the statistical information from the SL monolingual corpus. Thirdly, we will develop all possibilities of target result by applying dynamic programming technique. Finally, translated results of this system should be evaluated based on standard test set.

Acknowledgment

Special thanks to Dr. Krit KOSAWAT for helpful discussions, Mr. Prachya BOONKWAN and Mr. Taneth RAUNGRAJITPAKORN for insightful comments, and Mr. Sitthaa PHAHOLPHINYO and Ms. Monthika BORIBOON for providing the corpora. This research was supported in part by the National Electronics and Computer Technology Center (NECTEC), Thailand.

REFERENCES

[1] D.J. Arnold and Louis des Tombe, *Basic theory and methodology in Eurotra*. In Sergei Nirenberg, editor, Cambridge University Press, Cambridge, 1987, pp. 114-135.

[2] Peter et al Brown. "A statistical approach to language translation," in *Proceedings of the 12th COLING*, 1988, pp. 71-76.

[3] Makoto Nagao, "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle," *Artificial and Human Intelligence*, North Holland, 1984, pp. 173-180.

[4] NECTEC. "ParSit: Online English-Thai Machine Translation Service" [Online]. Available: http://www.suparsit.com

[5] AgentDict. [Online]. Available: http://www.agentdict.net

[6] Sitthaa Phaholphinyo, Teerapong Modhiran, Nattapol Kritsuthikul, and Thepchai Supnithi, "A Practical of Memory-based Approach for Improving Accuracy of MT," in *Proceedings of MT Summit X*, 2005, pp. 41-46.

[7] SWATH. Smart Word Analysis for Thai., http://www.links.nectec.or.th/download.php

[8] Donald R. Morrison, "PATRICIA-Practical Algorithm To Retrieve Information Coded in Alphanumeric," in ACM Journal, Vol. 15, No. 4, October 1968, pp. 514-534

[9] Toni Badia, Gemma Boleda, Maite Melero, and Antoni Oliver, "An *n*-gram approach to exploiting a monolingual corpus for Machine Translation," in *Proceedings EBMT Workshop of MT Summit X*, 2005, pp. 1-7.

[10] Peter Dirix, Ineke Schuurman, and and Vincent Vandeghinste, "METIS-II: Example-based machine translation using monolingual corpora – System description," in *Proceedings of EBMT Workshop of MT Summit X*, 2005, pp. 43-50.

[11] John Fry, "Assembling a parallel corpus from RSS news feeds," in *Proceedings of EBMT Workshop of MT Summit X*, 2005, pp. 59-62.

[12] John HUTCHINS, "Towards a definition of example-based machine translation," in *Proceedings of EBMT Workshop of MT Summit X*, 2005, pp. 63-70.

[13] Stella Markantonatou, Sokratis Sofianopoulos, Vassiliki Spilioti, Yiorgos Tambouratzis, Marina Vassiliou, Olga Yannoutsou, and Nikos Ioannou, "Monolingual Corpus-based MT using Chunks," in *Proceedings of EBMT Workshop of MT Summit X*, 2005, pp. 91-97.

[14] Vincent Vandeghinste, Peter Dirix, and Ineke Schuurman, "Example-based Translation without Parallel Corpora: First experiments on a prototype," in *Proceedings of EBMT Workshop of MT Summit X*, 2005, pp. 135-142.

[15] Satoshi Sato and Makoto Nagao, "Toward Memory based Translation," in *Proceedings of the 13th COLING*, 1990, pp. 247-252.

[16] Eiji Aramaki, Sadao Kurohashi, Satoshi Sato, and Hideo Watanabe, "Finding translation correspondences from parallel parsed corpus for example-based translation," in *Proceedings of MT Summit VIII*, 2001, 27-32.

[17] Taro Watanabe, and Eiichiro Sumita, "Example-based Decoding for Statistical Machine Translation," in *Proceedings of MT Summit IX*, 2003, pp. 410-417.

[18] Taro Watanabe, and Eiichiro Sumita, "Bidirectional decoding for statistical machine translation," in *Proceedings of 19th COLING*, 2002, pp. 1079-417.