Proceedings of the World Congress on Engineering 2012 Vol II
WCE 2012, July 4 - 6, 2012, London, U.K.

Machine Translation Based on Data Mining and Deductive Schemes

S. M. Fakhrahmad, A.R. Rezapour, M.H. Sadreddini, M. Zolghadri Jahromi

***Abstract*—Machine translation (MT) is one of the most attractive fields in natural language processing. In this paper, we propose some new ideas for designing an MT system. For this purpose, we first introduce a grammatical rule induction method. After representing the extracted knowledge by a set of finite automata, a recursive model is proposed, which uses a combination of rule and example based techniques. In the translation phase, through a hierarchical chunking process, the input sentence is divided into a set of phrases. Each phrase is first searched through the corpus of examples. If the phrase is found, it will not be chunked anymore. Otherwise, the phrase is divided into smaller sub-phrases. The experimental results show a promising accuracy and efficiency of the proposed system.**

***Index Terms*— Machine translation; Example-based; Rule-based; Corpora-based; Finite Automata; Grammar induction.**

I. INTRODUCTION

Machine Translation is one of the most attractive and applied fields in natural language processing (NLP). Machine translation (MT) is the process of automatically analyzing a text in a source language and producing the equivalent text in a target language. To date, machine translation has met with limited success. Conventional machine translation systems used to adopt *rule-based* (RBMT) methods, in which grammatical and linguistic restrictions are applied for translation. However, rule-based machine translation systems have many shortcomings. The major issues include ambiguity resolution and meaning interpretation. Rule-based systems suffer from inability to select the most suitable equivalent translation in many cases. Moreover, the rule-based systems are language-dependent since they are designed such that they can just be used for a specific pair of languages (source and target languages) [1-3].

In recent years, the mostly attended models of MT have been data-driven or corpus-based which is in sharp contrast to the dominant framework of the previous decades, i.e., RBMT. There are two corpora-based categories of translation methods namely, *example-based* (EBMT) and *statistics-based* (SMT) approaches proposed to overcome the shortcomings of rule-based methods [4]. In both cases the corpora comprise bilingual texts (original texts coupled with their translations) [5-8].

The EBMT approach is based on the extraction and combination of phrases (or other short segments of texts). In EBMT methods, a large set of translation samples (i.e., pairs of source text and its translation) are stored and used for similar translations. Example-based methods are mostly used in order to detect and translate expressions. The origin of EBMT can be dated precisely to a conference paper in 1984 by Makoto Nagao [9].

EBMT systems use segments (word sequences and not individual words) of source language texts extracted from a text corpus to build texts in a target language with the same meaning. The basic units for EBMT are sequences of words (phrases, or ‘fragments’), and the basic techniques are the matching of input phrases against sample source language phrases in the database, the extraction of corresponding target language phrases and the recombination of the segments as acceptable target language sentences.

The SMT approach was first proposed by Warren Weaver in 1949 [10]. It was then re-introduced in more details by researchers of IBM's Thomas J. Watson Research Center in 1991 [11]. This approach is primarily based on the study of frequencies of various linguistic units, including words, lexemes, morphemes, letters, etc., in a sample corpus to calculate a set of probabilities, so that various linguistic problems such as sense ambiguity can be solved. In other words, translation is based on statistical or probabilistic models whose parameters are extracted from the analysis of a bilingual corpus. Today, SMT methods are widely-studied and have attracted the attention of many other researchers in the field of machine translation [12-23].

Although EBMT and SMT techniques outperform rule-based methods in terms of translation accuracy, they still have their own problems. For example, both methods require a huge bilingual corpus containing all possible word combinations, which is hardly assured to be available. Moreover, RBMT methods are usually much faster than corpora-based methods, since they rarely need to perform interpretation and deduction tasks.

Indeed, in some cases, where we aim to have a more successful translation, making use of both RBMT and corpora-based techniques is inevitable. Another challenge is that there is really no efficient algorithm to extract knowledge from a large-scale corpus, which is required for ambiguity resolution and other related problems.

In this paper, we propose a new translation method

S.M. Fakhrahmad is with the Department of Computer Engineering, Islamic Azad University, Shiraz branch, Iran, (corresponding author, phone: +98-9177038028; e-mail: mfakhrahmad@ cse.shirazu.ac.ir).

A.R.Rezapour, M.H. Sadreddini and M.Zolghari Jahromi are with the Dept. of Computer Eng. and IT, Shiraz University, Iran (e-mail: {Rezapour, zjahromi, sadredin} @Shirazu.ac.ir).

called AMT[1], which can be considered as a hybrid of rule-based and corpora-based techniques. The rule-based part of the system is not language-dependent, since the grammatical rules are automatically generated from a large bi-lingual corpus. A set of novel schemes for knowledge representation as well as new methods of translation components (including hierarchical part-of-speech (POS) tagging, Automata construction, Automata traversing, etc) will be presented in this work. The rest of the paper is organized as follows. In Section 2, the whole structure of the proposed system will be presented. In the first part of this section, we illustrate the method we use to induce grammatical rules from a corpus and introduce some novel schemes for representation of the extracted knowledge. The rest of this section is devoted to introduction of the translation engine. In Section 3, we use some standard metrics to evaluate the system's accuracy and compare it with one of the well-known English-to-Persian translators.

II. The Proposed system

In this section, the proposed MT system (AMT) is introduced. AMT is composed of two main parts. The first part of our system performs grammar induction. Two main tasks are carried out in this part namely syntactic structure annotation and rule extraction. In this part, the grammatical rules of the source language and the translation order of the sentence chunks are induced from a large bi-lingual corpus. The discovered rules are represented in an automaton structure, which will then be used and traced by the translation engine. Translation engine is the main part of AMT. It uses a hybrid of RBMT and EBMT methods and uses a dictionary, a bilingual corpus and the set of extracted rules to translate and combine chunks of a sentence. In addition to the translation engine, a set of operations are required to complete and enhance the quality of the translation. They include chunking, stemming, sense disambiguation, discovery of the tense of the sentence (needed for the verb construction in the target language), etc.

A. Grammar Induction

This part of the system aims to discover and extract all possible syntactic structures of the source language by processing a large bi-lingual corpus. As will be discussed, two different structures, namely Finite automata and treebanks are simultaneously used to present and handle the syntactic schemes. The induced grammatical rules are presented in nested finite automaton structures, while the treebank structure is used to present syntactic annotated contexts. A treebank is a parsed corpus in which each sentence has been parsed and annotated with syntactic structure. Since the syntactic structure is represented as a tree, it is called as treebank. The alternative term Parsed Corpus is sometimes used for treebank. There are two main categories of treebanks: treebanks that annotate phrase structure (such as Penn Treebank [24]) and those that annotate dependency structure (such as the Quranic Arabic Dependency Treebank [25]).

In order to build a treebank, each sentence has to be annotated with syntactic structure. This can be carried out manually by linguists or semi-automatically, where a parser performs the annotation task. In the second case, linguists usually have to check and correct the result, which can be very labor intensive depending on the level of annotation details we want to present.

Figure 1 shows the annotated scheme, for the example sentence "The sun sets in the west".

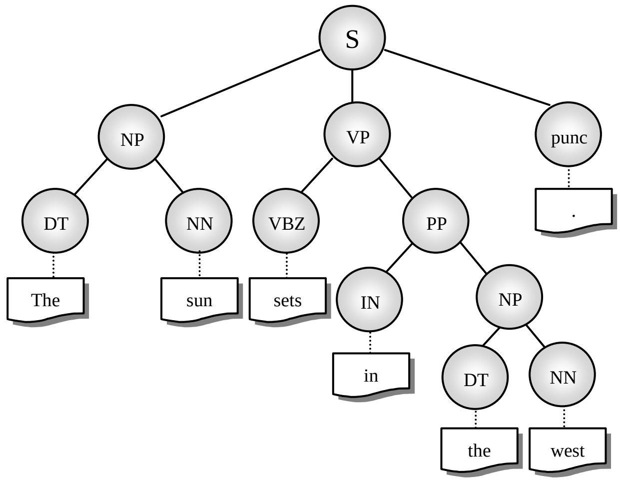

Fig 1. The tree structure

After performing syntactic structure annotation on all contexts, each sentence is divided into a set of phrases, where each phrase is composed of a set of words or smaller phrases. By processing and comparing the set of all phrase sequences, we can find the sequences that are frequent Thus, they should all be extracted and recorded for further use. Since the order of parts of a discovered sequence will often be changed after translation (in the target language), a word alignment process has to be carried out as a complementary process on the bilingual corpus. That is, the translation order of each item of a discovered sequence is recorded so as to be used for further translation purposes.

B. Constructing Finite Automata

In this stage, for each of the main phrases (i.e., S, VP, NP, ADJP, etc) all the phrase sequences discovered in the previous section are integrated to constitute a finite automaton. Finite automaton is one of the major components used in the translation engine. Every sequence discovered before is represented as a path in automata which connects the start state of the automaton to a final state (maybe passing through a set of middle states). Different POS tags can be seen as the labels of transitions within the automata. Each transition label is coupled with its translation order (in target language) which had been obtained through the alignment process in the last part. The input of the automaton (when being used for translation) is a parsed and POS tagged sentence which enters phrase (or word) by phrase (or word). Each phrase changes the current state of the automaton. Thus, the input sentence traverses a path through the automaton, which will specify the translation pattern.

Figure 2 shows the structure of the finite automaton built

[1] Automata-based Machine Translator

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

to represent different possible structures of a sentence.

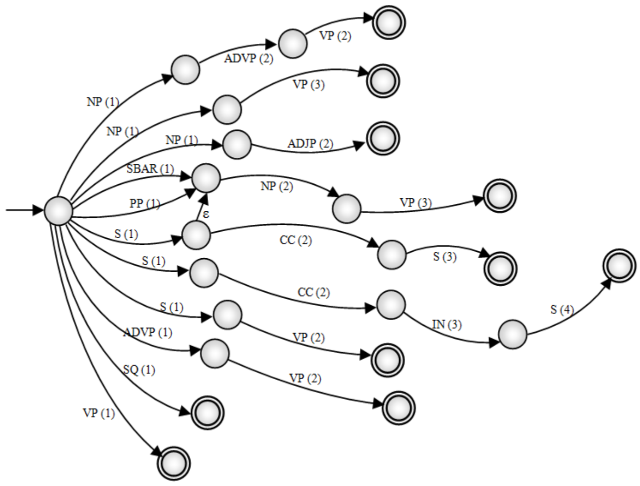

Fig 2. The finite automaton structure representing possible structures for a sentence in English

Although for each structure a separate automaton is constructed, the automaton structures are not independent. Indeed, the set of finite automaton structures are tightly coupled in a nested scheme.

C. Translation engine

The next part of AMT is the translation engine. the translation engine uses the parsed bi-lingual corpus as well as the set of automata constructed in the grammar induction phase. The translation method we use in this part includes both rule-based and corpora-based strategies. In the proposed method, each sentence or phrase is first searched through the set of examples in the corpus. If it is found in the corpus, its aligned meaning will directly be used in the translation. Otherwise, the sentence or phrase will iteratively be divided into smaller parts using the same chunking process performed in the previous part. The chunks are then fed to the related automaton to find the translation rule. As discussed in Section 2-2, if one of the chunks is a phrase (rather than a simple word), its own automaton has to be traversed. Thus, in such cases, traversing of the current automaton is stopped temporarily until the nested automaton is finished (just the same as procedure calls in software programs).

The best case in this system occurs when a sentence is found in the corpus of examples. In this case, no chunking process is required and the translation type is completely example-based. On the other hand, the worst case occurs when neither the whole sentence nor its parts could be found in the corpus. In this case, the chunking process continues hierarchically until the sentence is decomposed into to a set of words. In this case, the translation will be fully rule-based. Thus, the order of translation of the words has to be found from the set of previously built automata. Since through the chunking process, the words are placed in different branches of a tree in different depths, the order found for each part is recorded in a form that can represent its situation in the tree, too. Figure 3 presents the main function of the translation engine. The function *Translate* receives a sentence (or a part of a sentence) as an XML node. The output is a sorted list of items in target language which represents the primary translation of the input sentence. In order to translate nested phrases, the function is invoked recursively whenever a phrase is met. The *current-state* parameter shows the state of the automaton which is being traversed. For each call of the Translate function, *current-state* is initialized to ‘Start’. Each part of the input text changes the value of *current-state* as it is visited.

```
Function Translate
inputs:    text: XmlNode,    current-state: String
Output:    meaning-list: List of the chunks’ meanings
{
For each XmlNode such as node in child-nodes of text
    If type of node is ‘chunk’
        Look in corpus to see if the chunk exists in the corpus
        If the chunk was found in corpus
            Compute its translation order according to automata
            Add the chunk and its translation to the meaning-list
        Else
            Translate(node, ‘Start’);
    Else if type of node is ‘word’
        Find the meaning of the word from dictionary
    Change the current-state of automata  according to the value
of the Transition attribute of node;
If type of current-state is ‘Final’
    Sort items in meaning-list according to their translation
order values
}
```

Fig 3. The main procedure of the translation engine

III. Evaluation

Evaluation of machine translation output is a challenging problem. Automatic evaluation is simply defined as the comparison of the actual translation performed by the system (denoted as *candidate translation*) and the desired translation (denoted as *reference translation*). However, in most cases, there is not just a single correct translation for an input sentence. In the extreme case, a pair of translations for the same input can be perfectly valid, while they have different structures and include completely different words. There are many evaluation metrics already proposed, all of which assign a score to the translation output. In this work, we consider some of these evaluation metrics, which are most commonly used and most suitable to perform the experiments. Before presenting the experiments, we first give a brief description on some of the common evaluation measures.

A. Common evaluation measures

Word Error Rate (WER) [28]

Most This metric represents the dissimilarity or distance of a pair of translations via the Edit distance measure. The Edit distance is defined as the minimum number of word insertions, substitutions and deletions required to convert the candidate translation into the reference translation. All

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

three operations are assumed to have the same costs.

The value of WER is measured by dividing the number of edit operations by the number of words in the reference translation. If the candidate translation is longer than the reference, the value of WER will be greater than 1. Thus, WER has a bias towards shorter hypotheses.

When there is more than one reference translation, the reported error (WER) for a candidate translation is the minimum error over all references.

Position-independent Word Error Rate (PER) [29]

Unlike WER that requires exactly the same order of the words in candidate and reference translations, PER neglects word order, absolutely. It measures the difference of the words occurring in candidate and reference translations. The resulting number is then divided by the number of words in the reference translation.

Translation Edit Rate (TER) [30]

TER is another error measure that counts the number of edits required to convert a system output into one of the given references. This metric can measure the amount of human work that would be required to post-edit the translations proposed by the system and convert to the reference translation. In contrast to WER, movements of blocks are permitted and counted as one edit with equal costs to other legal operations, i.e., insertions, deletions and substitutions of single words.

The value of TER is obtained by dividing the number of edit operations by the average number of reference words.

BLEU [31]

The BLEU is one of the most well-known metrics which is frequently used in evaluation of translation systems. In contrast to other metrics defined above, BLEU is a precision (or similarity) metric. It measures the similarity of n-gram vectors in the reference translations and the candidate translation. In other words, it represents the rate of n-grams of the candidate translation, which can also be found in the reference translation. If more than one reference exists, the counts are gathered for all translations.

Since BLEU is a precision measure, higher values indicate better results. If no n-gram of maximum length matches between candidate and reference translations, the BLEU score will be zero.

NIST [32]

NIST is another precision measure which is considered as an improved version of BLEU. When using this measure, n-gram occurrences are weighted by their importance. The importance of an n-gram is specified according to the frequency of the n-gram in the reference translations.

NIST considers less importance values for frequently occurring n-grams in comparison with rare ones.

B. Evaluation Results

In the first part of the experiment, we used the BLEU score in order to evaluate the translation precision. For this purpose, we used a fraction of our bi-lingual corpus including 100 pairs of sentences. In order to obtain more reliable results, we divided the set of sentences into 5 blocks of 20 sentences each, and computed the BLEU metric considering 4-grams on these blocks individually. We thus have 5 samples of the BLEU metric for each system. We computed the means and variances, which are shown in Table 1.

In this experiment, the BLEU score was measured in two ways. The First case was the usual case where we considered the own words included in n-grams in order to measure the BLEU scores. In the second case, we used the POS tags of the words instead of the own words. The results of these two cases as well as the average value are given in Table 2.

TABLE I
THE EVALUATION RESULTS (MEAN AND VARIANCE) FOR TRANSLATION SYSTEMS ON 5 BLOCKS OF THE TEST CORPUS

	AMT
BLEU score (Mean)	0.564
Standard Deviation	0.018

TABLE II
BLEU SCORES RECEIVED BY TRANSLATION SYSTEMS IN TWO CASES

	AMT
BLEU Score (considering the words)	0.564
BLEU Score (considering the POS tags)	0.895
Average	0.729

IV. Conclusion

In this paper, we first proposed a grammar induction method. After representing the extracted knowledge in form of nested finite automata, a recursive model was proposed, which used a combination of rule and example based techniques. In the translation phase, through a hierarchical chunking process, the input sentence is divided into a set of phrases. Each phrase is searched in the corpus of examples. If the phrase is found, it will not be chunked anymore. Otherwise, the phrase is divided into smaller sub-phrases. The worst case occurs when none of the phrases and sub-phrases can be found in the corpus. In this case, we will finally have a set of simple words and the translation procedure will completely be rule-based. In other cases both approaches are applied. The accuracy of the system in translating from English to Persian was evaluated through a set of experiments using various metrics. The simulation results showed the promising accuracy and efficiency of the proposed system.

References

[1] R.M.K. Sinha and A. Jain, AnglaHindi: An English to Hindi Machine Translation System, MT Summit IX, New Orleans, USA, Sept.23-27, 2003, pp. 112-119.

[2] S. Dave, J. Parikh and P. Bhattacharyaa. Interlinguabased English-Hindi Machine Translation and Language Divergence. Machine Translation 16(4), 2001, pp. 251-304.

[3] M. Nagao, A framework of a mechanical translation between Japanese and English by analogy principle. In: A.Elithorn and R.Banerji (eds.) Artificial and human intelligence (Amsterdam: North-Holland), 1984, pp.173-180.

[4] R. Navigli, P. Velardi, Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 27(7), 2005, pp. 1063–1074.

[5] C. Lavecchia, K. Smaili and D. Langlois: Building parallel corpora from movies, in Proceedings of the 4th International Workshop on Natural Language Processing and Cognitive Science, (Funchal, Madeira, 12-13 June 2007), no pagination, 2007.

[6] M. Carl, A. Way, Introduction to special issue on example-based machine translation, Machine Translation, Volume: 19, (3-4), 2007, pp. 193-195

[7] J. Hutchins, Example-based machine translation: a review and commentary. Machine Translation vol.19, 2005, pp. 197-211.

[8] E. Sumita and H. Iida, Experiments and prospects of Example-Based Machine Translation, ACL '91 Proceedings of the 29th annual meeting on Association for Computational Linguistics, 1991, pp 185-192

[9] M. Nagao, T. Nishida and J. Tsujii, Dealing with incompleteness of linguistic knowledge in language translation – transfer and generation stage of Mu machine translation project. Coling84: 10th International Conference on Computational Linguistics & 22nd Annual Meeting of the Association for Computational Linguistics, Stanford University, California. Proceedings, 1984, pp.420-427.

[10] W. Weaver, Translation. Repr. in: Locke, W.N. and Booth, A.D. (eds.), Machine translation of languages: fourteen essays (Cambridge, Mass.: Technology Press of the Massachusetts Institute of Technology, 1955), 1949, pp. 15-23.

[11] IBM's Thomas J. Watson Research Center: http://www. watson.ibm.com

[12] S. Abdul-Rauf and H. Schwenk, On the use of comparable corpora to improve SMT performance. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), 2009, pp. 16-23.

[13] O. Bojar and A. Tamchyna, Forms Wanted: Training SMT on Monolingual Data. Abstract at Machine Translation and Morphologically-Rich Languages. Research Workshop of the Israel Science Foundation University of Haifa, Israel, 2011.

[14] D. Chiang, Hierarchical phrase-based translation. Computational Linguistics, 33(2), 2011, 201–228.

[15] N. Habash, Four techniques for online handling of out-of-vocabulary words in Arabic-English statistical machine translation. In ACL 08, 2008, pp. 77-86.

[16] Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 2007, pp. 967-975.

[17] P. Koehn, F. J. Och, and D. Marcu, Statistical phrased-based machine translation. In HLT/NACL, 2003, pp. 127–133.

[18] 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 ACL, demonstration session, 2007.

[19] F. J. Och and H. Ney, Discriminative training and maximum entropy models for statistical machine translation. In Proc. of the Annual Meeting of the Association for Computational Linguistics, 2002, pages 295–302.

[20] F. J. Och and H. Ney, A systematic comparison of various statistical alignement models. Computational Linguistics, 29(1), 2003, pp.19–51.

[21] H. Schwenk, Investigations on largescale lightly-supervised training for statistical machine translation. In IWSLT, 2008, pp. 182–189.

[22] M. Banko and R. C. Moore. Part of speech tagging in context. In Proceedings of the Inter-national Conference on Computational Linguistics (COLING), 2004, pp. 164-170.

[23] Goldwater and T. L. Griffiths, A fully Bayesian approach to unsupervised part-of-speech tagging. In Proceedings of the ACL, 2007, pp.187-192.

[24] A corpus of parsed sentences. Used by many researchers for training data-driven parsing algorithms: www.ldc.upenn.edu/ ldc/online/treebank

[25] K. Dukes, T. Buckwalter, A Dependency Treebank of the Quran using traditional Arabic grammar, Informatics and Systems (INFOS), 2010 , pp. 1-7.

[26] G. McLachlan and T. Krishnan. The EM Algorithm and Extensions. John Wiley & Sons, New York, 1996.

[27] Goldberg, M. Adler, and M. Elhadad. EM can find pretty good HMM POS-taggers (when given a good start). In Proceedings of the ACL, 2008.

[28] G. Chiorboli, B. De Salvo, G. Franco and C. Morandi, Some thoughts on the word error rate measurement of A/D converters, IEEE International Conference on Electronics, Circuits and Systems, Vol. 3, 1998, pp. 453 – 456.

[29] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf, Accelerated DP based search for statistical translation. In Fifth European Conf. on Speech Communication and Technology, Rhodos, Greece, September 1997, pp. 2667–2670.

[30] M. Snover, B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul, A study of translation edit rate with targeted human annotation. In Proceedings of Association for Machine Translation in the Americas, 2006.

[31] Kishore Papineni, Salim Roukos, ToddWard, andWie-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proc. of the 40th AnnualMeeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, 2002, pp. 311–318.

[32] G. Doddington, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proc. ARPAWorkshop on Human Language Technology, 2002.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)