Hierarchical Markov based Reordering Model for English-Myanmar Translation

Thinn Thinn Wai
University of Computer Studies, Yangon, Myanmar
thin2wai@gmail.com

Abstract

In statistical machine translation, reordering is crucial component for translation of the different languages with different word orders. Without reordering during language translation, sentences can only be translated properly into a language with similar word order. An effective reordering scheme is essential to model translation between languages with different word orders, such as SVO-languages (English or Chinese) and SOV-languages (Japanese or Turkish or Myanmar). In this paper, we focus on reorder scheme for lexical reordering to incorporate into our translation model. If a sentence is given, our proposed reordering system performs three tasks on this sentence such as word level reordering, chunk level reordering and clause level reordering. In this research, Markov based Reordering Model will be implemented. Moreover, the movement parameters to input the reordering model are taken from the rules automatically generated from parallel aligned corpus.

1. Introduction

In machine translation (MT), one of the main problems to handle is word reordering. A word is "reordered" when it and its translation occupy different positions within the corresponding sentence. In Statistical MT (SMT) (Brown et al., 1993), word reordering is faced from two points of view: constraints and modeling. If arbitrary word reorderings are permitted, the exact decoding problem is NP-hard (Knight, 1999); it can be made polynomial time by introducing proper constraints, such as IBM constraints (Berger et al., 1996) and Inversion Transduction Grammars (ITG) constraints (Wu, 1997). Among all the allowed word-reordering, it is expected that some are more likely than others. The aim of reordering models, known also as distortion models, is that of providing a measure of the plausibility of word movements. Most of the distortion models developed so far is unable to exploit linguistic context to score reordering: they just predict target positions on the basis of other (source and target) positions.

In this work we present a novel word reordering model. In particular, our goal is to model reordering concerning three levels: word level, chunk level and clause level and will use function tag and pos (part of speech) tag information to extract reordering rules.

And then first order Markov model is used to move words in source language order to target language order based on the movement parameter extracted from reordering rules. Moreover, lexical information is used to disambiguate the same pattern rules in this reordering model. Relevant statistics are collected from word aligned parallel texts/corpus regarding the distance between target words and the distance between the corresponding source words.

2. Related Work

Different approaches have been developed to deal with the word order problem. First approaches worked by constraining reordering at decoding time (Berger et al., 1996). In (Wu, 1996), the alignment model already introduces restrictions in word order, which leads also to restrictions at decoding time. A comparison of these two approaches can be found in (Zens and Ney, 2003). They have in common that they do not use any syntactic or lexical information; therefore they rely on a strong language model or on long phrases to get the right word order. Other approaches were introduced that use more linguistic knowledge, for example the use of bitext grammars that allow parsing the source and target language (Wu, 1997). In (Shen et al., 2004) and (Och et al., 2004) syntactic information was used to re rank the output of a translation system with the idea of accounting for different reordering at this stage. In (Tillmann and Zhang, 2005) and (Koehn et al., 2005) a lexicalized block-oriented reordering model is proposed that decides for a given phrase whether the next phrase should be oriented to its left or right.

The most recent and very promising approaches that have been demonstrated, reorder the source sentences based on rules learned from an aligned training corpus with a POS-tagged source side (Chen et al., 2006), (Popovic and Ney, 2006) and

(Crego and Marino, 2006). These rules are then used to reorder the word sequence in the most likely way.

In our approach we follow the idea proposed in (Crego and Marino, 2006) of using a parallel training corpus with a tagged source side to extract rules which allow a reordering before the translation task. As a new feature we use the context in which a reordering pattern is seen in the training data. Context refers to the words or tags to the left or to the right of the sequence for which a reordering has been observed. By doing this we hope to differentiate between reorderings that are dependent on their context.

3. Statistical Machine Translation

The goal of machine translation is the translation of an input string s1; : : : ; sJ in the source language into a target language string t1; : : : ; tI . We choose the string that has maximal probability given the source string $\Pr(t_1^I / s_1^J)$. Applying Bayes' decision rule yields the following criterion:

$$\arg\max_{t_1^I} \Pr(t_1^I / s_1^J)$$

$$= \arg\max_{t_1^I} \{\Pr(t_1^I).\Pr(s_1^J / t_1^I)\} \quad (1)$$

Through this decomposition of the probability, we obtain two knowledge sources: the translation and the language model. Those two can be modeled independently of each other. The correspondence between the words in the source and the target string is described by alignments that assign target word positions to each source word position. The probability of a certain target language word to occur in the target string is assumed to depend basically only on the source words aligned to it. The search is denoted by the argmax operation in Eq. 1, i.e. it explores the space of all possible target language strings and all possible alignments between the source and the target language strings to find the one with maximal probability. The input string can be preprocessed before being passed to the search algorithm. If necessary, the inverse of these transformations will be applied to the generated output string.

For descriptions of SMT systems see for example (Germann et al., 2001; Och et al., 1999; Tillmann and Ney, 2002; Vogel et al., 2000;Wang andWaibel, 1997).

4. System overview

In this reordering system, there are five processes: sentence collection, function tagging and Pos tagging, corpus creation, reordering rule extraction, and reordering using the extracted rules. For function tagging, language analyzer described in [15] is use. For pos tagging, "TreeTagger" is used. And then English-Myanmar bilingual corpus is created using information resulted from analyzer and alignment model. After creation corpus, reordering rules are automatically generated using parallel aligned corpus. And then reordering is performed using extracted reordering rule.

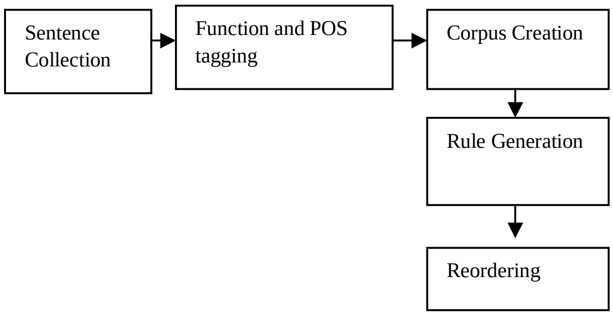

Figure 1: Overview of the system

5. Reordering in English-Myanmar Translation

The chunk order of English and Myanmar is different in reading after transferring English text to Myanmar text since English is SVO language and Myanmar is an SOV language. More precisely Myanmar is a verb-final language. There are many phenomena which lead to differing word order between English and Myanmar. Two of them is negation (the differing placement of items such as not in English and "မ" in Myanmar), and also verb particle constructions. Moreover, Myanmar syntax has several characteristics that lead to significantly different word order from that of English. Another example concerns verb-particle constructions. Without the reordering, the particle can be arbitrarily far from the verb that it modifies and there is a danger in the natural translation when no particle is present. Moreover, Myanmar is modifier and adjunct proceeding language [13]. In our translation, the relationships between chunks and the order in which chunks occur are much flexible for reordering after transferring source to target language. For a given source text sentence, we will reorder at three different levels; words level where words in each chunk are reordered in target language word order, chunks level where chunks in specific sentence clauses are reordered and clauses level where clauses in the sentence are reordered. It is also needed to complement some particle words to make raw Myanmar text smooth. This task can be done by adding appropriate words between chunks since the places to add these words are at chunk boundaries. And the words to be added can also be decided based on function tag of the chunk.

6. Proposed Reordering Model

In this reordering scheme we proposed, there are three main processes. The first process is to reorder the input language text clauses into target language clause order. The second process is to reorder the phrases in each clause to target language phrase order by chunk to chunk. The third process is to reorder the words in the chunk that conform to the target language sentence pattern. All these processes will be performed by using First Order Markov reorder model. As a first step in the translation process, we analyzed and tag each element of the input sentence text using language analyzer. The second step translates the words and phrases of source language into target language. The third step is to reorder the English sentence depending on the chunk function structure. The English input sentence e is segmented into a sequence of n phrases (e1,e2,…………,en). We assume a uniform probability distribution over all possible segmentations. Each English phrase $e_i^{\sim}$ in this sequence is translated into a Myanmar phrase $b_i^{\sim}$. Reordering of the Myanmar output phrases is modeled by a relative distortion probability distribution d $(a_i - b_{i-1})$. Phrase reordering takes as its input a English phrase sequence in English phrase order $x_1, x_2, ..., x_k$. This is then reordered into Myanmar phrase order $g_1, g_2, ..., g_k$. Given an input phrase sequence $^{x}K_1$ we now associate a unique jump sequence $^{m}K_1$ with each permissible output phrase sequence $^{g}K_1$. The jump m_k measures the displacement of the k^{th} phrase x_k. The jump sequence $^{b}K_1$ is constructed such that $^{g}K_1$ is a permutation of $^{x}K_1$. Each jump b_k depends on the phrase-pair (x_k, g_k) and preceding jumps b_{k-1}. As such Markov process model permutes input phrases (X_1, X_2, X_{13}………….. X_n) into phrase sequence (Y_1, Y_2, Y_3………….. Y_n) and the bilingual rule analyzer checks the sentence structure is conformed. This model accepts the parameter that controls the degree of movement. The phrase alignment sequence specifies a reordering of phrases into predefined phrase order; the words within the phrases remain in the source language order.

$$P(m_1^K \setminus x_1^K, K, g_1^K) = P(m_1^K \setminus x_1^K)$$

$$= \prod_{k=1}^{K} P(m_k \setminus x_k, \Phi_{k-1}) \quad (2)$$

Where $g_1,g_2,\ldots,g_k$ is the predefined sentence structure, Φ_{k-1} is the state arrived by m_1^{k-1} and m_k is movement parameter for phrase x_k. And so y_k is determined by x_k and m_k. Displacement of k^{th} phrase x_k is $x_k \rightarrow y_k + m_k, k \in \{1,2,...,K\}$

The jump sequence m_1^K is constructed such that y_1^K is the permutation of x_1^K. The simple ways to swap phrases are:

(i) Adjacent phrases (jump 1 phrase)

(ii) Within three phrases (jump 2 phrase)

Figure 2 and 3 show the values of movement parameter when make a swapping.

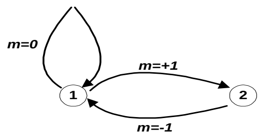

Figure 2: Jumping 1 phrase (adjacent)

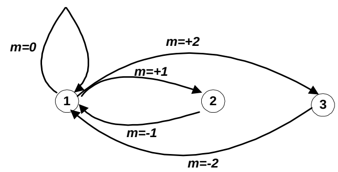

Figure 3: Jumping 2 phrases (three phrases window)

Now we use jumping 1 phrase method for reordering. Figure 4 illustrates reordering steps to make input chunk sequence conformed to the corresponding target language sentence structure. For using 1 phrase jumping method, the value allowed for movement parameter m_k is $m_k \in \{-1, 0, 1\}$. And which has two equivalence states for any history $m_1^{k-1}; \phi(m_1^{k-1}) \in \{1,2\}$. A jump of +1 has to be followed by a jump of -1, and 1 is the start and end state and so $\sum_{k=1}^{K} m_k = 0$.

7. Reordering Rules

In order to overcome the limitations of our simple distortion model, we propose to enrich the translation process with reordering rules defined as in the following. Function tag-based on which rules are composed of function tags. These Function tag rules are used in clause level and chunk level reordering. POS-based on which rules are composed of part of speech (POS) tags and used in word level reordering.

7.1. Definition of Reordering Rule

A reordering rule consists of two sides: the left-hand-side (lhs), which is a function tags or POS tags pattern, and the right-hand-side (rhs), which corresponds to a possible reordering of that pattern. Different rules can share the lhs: in such cases, the same pattern can be reordered in more than one way. Rules are weighted, according to statistics extracted from training data. There are two kinds of reordering patterns: function-based, which defines reordering at the clause and phrase level, and pos-based, which defines reordering at the word level. Let us consider the following examples:

- **Rules using function tag**

-SUBJ, ACTIVE, OBJ#0/0, 1/2, 2/1:7(10)
-SUBJ, ACTIVE, OBJ#0/1, 1/2, 2/0:3(10)
-SUBJ, ACTIVE, ADVL# 0/0, 1/2, 2/1:10(10)
-F-SUBJ, ACTIVE, PCOMPL-S, ADVL, OBJ-P# 0-1/3, 2/2, 3/1, 4/0:10(10)

- **Rules using pos tag**

-DT, JJ, NN#1/0, 2/1, 0/2
-a:: DT, NN#0/1, 1/0:4(12)
-this:: DT, NN#0/0, 1/1:4(12)
-the:: DT, NN#0-1/0:4(12)
-CD, NNS#1/0, 0, 1:10(10)

In the above rules, "SUBJ","ACTIVE" and OBJ are function tags and "DT","JJ","NN","CD","NNS" are POS's tags. Therefore, "SUBJ, ACTIVE, OBJ" is function rule pattern and "DT, JJ, NN" is POS rule pattern. The string of numbers after "#" is position of source and target words and source word position is divided by "/" target position. For example, in the rhs of the pos rule pattern"1/0, 2/1, 0,2", the 1/0 means that the pos tag at the position 1,"JJ" is move to the position 0. In this model, we used array structure to store the position and so the starting index is 0. Moreover, in the function tag rule, the formal subject(F-SUBJ) *"there"* in English is not in the Myanmar Function tag and then it is translated as "ရှိသည်" in Myanmar language by combining it into the Function tag(ACTIVE) containing the words *"am ,is are, was, were"*. This means that the two function tags F-SUBJ and ACTIVE are become only ACTIVE and F-SUBJ is dismissed in Myanmar. Therefore, in the third function tag rule described above, the string after #, "0-1/3" means that the words at position "0"and "1" are move together into the position "3".

The sequences "SUBJ, ACTIVE, OBJ" and "DT, JJ, NN" are function and pos rule patterns (p_1^n).The strings of numbers in between the symbols "#" and ":" represent suggested reordering (r_1^n): each integer after "/", r_i represents the new position of (the translation of) p_i. The two numbers after the colon (:) are collected from training data and are respectively the number of times the rhs (reordering suggestion) of the rule has been observed $count\ (r_1^n)$ and (inside brackets) the number of occurrences of the rule pattern $count\ (p_1^n)$. The probability of each reordering suggestion is computed as:

$$P(r_1^n / p_1^n) = \frac{count\ (r_1^n)}{count\ (p_1^n)} \tag{3}$$

And then, for the same pattern rule with the same probability, we use the context information (words) that is added before the rule pattern divided by ":: " to obtain the correct order.

8. Conclusions

We presented a reordering model based on rules learned from a tagged aligned corpus. The results we obtain show that this approach outperforms our previous word reordering strategy, which used only distance information. Moreover, we use array structure representations for tokens, POS tags and chunks. And then, tree representation is not used and it is sure that output phrases sequences are valid permutation of input phrase sequence. The model can make use of syntactic information and performs better for language pairs with different word orders and case marking schema. Our approach makes use of syntactic knowledge to overcome a weakness of tradition SMT systems, namely long-distance reordering.

9. References

[1] C. Tillmann and H. Ney. 2002. Word reordering and DP beam search for statistical machine translation to appear in Computational Linguistics.

[2] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast decoding and optimal decoding for machine translation. In Proc. 39th Annual Meeting of the Assoc. for Computational Linguistics joint with EACL, pages 228–235, Toulouse, France, July.

[3] S. Vogel, F.J. Och, C. Tillmann, S. Nießen, H. Sawaf, and H. Ney. 2000. Statistical methods for machine translation. InW.Wahlster, editor, Verbmobil: Foundations of Speech-to-Speech Translation, pages 377–393. Springer Verlag: Berlin, Heidelberg, New York.

[4] Y.Y. Wang and A. Waibel. 1997. Decoding algorithm in statistical translation. In Proc. 35th Annual Meeting of the Assoc. for Computational Linguistics, pages 366–372, Madrid, Spain, July.

[5] Ei Ei Han and Ni Lar Thein, "Morphological Synthesis For Myanmar Language", Proceeding of International Conference on Internet Information Retrieval, Korea, 2007.

[6] Yaser Al-Onaizan and Kishore Papineno. 2006. Distortion models for statistical machine translation. In Proceedings of the 21st International Conference on Computational Linguistics and the 4th annual meeting of the ACL, pages 529–536, Sydney, Australia.

[7] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra,1996. A maximum entropy approach to natural language processing. Computational Linguistics, 22(1):39.

[8] B. Chen, M. Cettolo, and M. Federico. 2006. Reordering rules for phrase-based statistical machine translation. In Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation, pages 1–15.

[9] M. Popovic and H. Ney. 2006. POS-based word reorderings for statistical machine translation. In Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC), page 1278, Genoa, Italy.

[10] L. Shen, A. Sarkar, and F. J. Och. 2004. Discriminative reranking for machine translation. In HLTNAACL 2004: Main Proc., page 177.

[11] C. Tillmann and T. Zhang. 2005. A localized prediction model for statistical machine translation. In Proceedings of the 43rd Annual Meeting of the As-soc. for Computational Linguistics (ACL), pages 557–564, Ann Arbor, MI.

[12] D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics, page 152.

[13] D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377.

[14] Y. Zhang, R. Zens, and H. Ney. 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST), pages 1–8, Rochester, NY.

[15] Myat Thuzar Tun and Ni Lar Thein, " English Syntax Analyzer for English-to-Myanmar Machine Translation", In proceedings of the Fifth International Conference on Computer Application, Myanmar, February, 8-9,2007.

[16] Myat Thuzar Tun, Tin Myat Htwe and Ni Lar Thein, "EMTM: An Effective Language Translation Model", In proceedings of International Conference on Internet Information Retrieval, Korea, November 30, 2005.

[17] Shankar Kumar “Local Phrase Reordering Models for Statistical Machine Translation”, Center for Language and Speech Processing, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, U.S.A.

[18] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, “The Mathematics of Statistical Machine Translation: Parameter Estimation,” Computational Linguistics, vol. 19(2), pp. 263–312, 1993.