A Structured Prediction Approach for Statistical Machine Translation

Dakun Zhang* **Le Sun†** **Wenbo Li***

*Institute of Software, Graduate University
Chinese Academy of Sciences
Beijing, China, 100080
{dakun04,liwenbo02}@iscas.cn

†Institute of Software
Chinese Academy of Sciences
Beijing, China, 100080
sunle@iscas.cn

Abstract

We propose a new formally syntax-based method for statistical machine translation. Transductions between parsing trees are transformed into a problem of sequence tagging, which is then tackled by a search-based structured prediction method. This allows us to automatically acquire translation knowledge from a parallel corpus without the need of complex linguistic parsing. This method can achieve comparable results with phrase-based method (like Pharaoh), however, only about ten percent number of translation table is used. Experiments show that the structured prediction approach for SMT is promising for its strong ability at combining words.

1 Introduction

Statistical Machine Translation (SMT) is attracting more attentions than rule-based and example-based methods because of the availability of large training corpora and automatic techniques. However, rich language structure is difficult to be integrated in the current SMT framework. Most of the SMT approaches integrating syntactic structures are based on probabilistic tree transducers (tree-to-tree model). This leads to a large increase in the model complexity (Yamada and Knight 2001; Yamada and Knight 2002; Gildea 2003; Galley et al. 2004; Knight and Graehl 2005; Liu et al. 2006). However, formally syntax-based methods propose simple but efficient ways to parse and translate sentences (Wu 1997; Chiang 2005).

In this paper, we propose a new model of SMT by using structured prediction to perform tree-to-tree transductions. This model is inspired by Sagae and Lavie (2005), in which a stack-based representation of monolingual parsing trees is used. Our contributions lie in the extension of this representation to bilingual parsing trees based on ITGs and in the use of a structured prediction method, called SEARN (Daumé III et al. 2007), to predict parsing structures.

Furthermore, in order to facilitate the use of structured prediction method, we perform another transformation from ITG-like trees to label sequence with the grouping of stack operations. Then the structure preserving problem in translation is transferred to a structured prediction one tackled by sequence labeling method such as in Part-of-Speech (POS) tagging. This transformation can be performed automatically without complex linguistic information. At last, a modified search process integrating structure information is performed to produce sentence translation. Figure 1 illustrates the process flow of our model. Besides, the phrase extraction is constrained by ITGs. Therefore, in this model, most units are word based except that we regard those complex word alignments as a whole (i.e. phrase) for the simplicity of ITG-like tree representations.

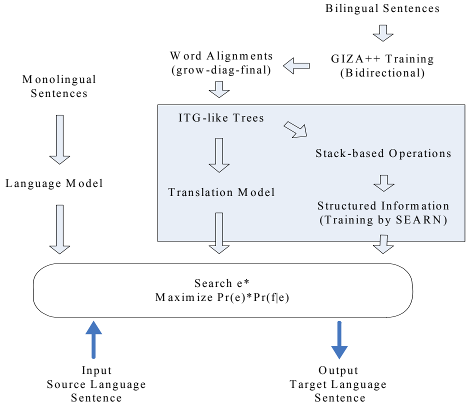


Figure 1: Chart of model framework

The paper is organized as follows: related work is show in section 2. The details of the transformation from word alignments to structured parsing trees and then to label sequence are given in section 3. The structured prediction method is described in section 4. In section 5, a beam search decoder with structured information is described. Experiments are given for three European language pairs in section 6 and we conclude our paper with some discussions.

2 Related Work

This method is similar to block-orientation modeling (Tillmann and Zhang 2005) and maximum entropy based phrase reordering model (Xiong et al. 2006), in which local orientations (left/right) of phrase pairs (blocks) are learned via MaxEnt classifiers. However, we assign shift/reduce labeling of ITGs taken from the shift-reduce parsing, and classifier is learned via SEARN. This paper is more elaborated by assigning detailed stack-operations.

The use of structured prediction to SMT is also investigated by (Liang et al. 2006; Tillmann and Zhang 2006; Watanabe et al. 2007). In contrast, we use SEARN to estimate one bilingual parsing tree for each sentence pair from its word correspondences. As a consequence, the generation of target language sentences is assisted by this structured information.

Turian et al. (2006) propose a purely discriminative learning method for parsing and translation with tree structured models. The word alignments and English parse tree were fed into the GenPar system (Burbank et al. 2005) to produce binarized tree alignments. In our method, we predict tree structures from word alignments through several transformations without involving parser and/or tree alignments.

3 Transformation

3.1 Word Alignments and ITG-like Tree

First, following Koehn et al. (2003), bilingual sentences are trained by GIZA++ (Och and Ney 2003) in two directions (from source to target and target to source). Then, two resulting alignments are recombined to form a whole according to heuristic rules, e.g. grow-diag-final. Second, based on the word alignment matrix, one unique parsing tree can be generated according to ITG constraints where the “left-first” constraint is posed. That is to say, we always make the leaf nodes as the right sons as possible as they can. Here we present two basic operations for mapping tree items, one is in order and the other is in reverse order (see Figure 2). Basic word alignments are in (a), while (b) is their corresponding alignment matrix. They can be described using ITG-like trees (c).

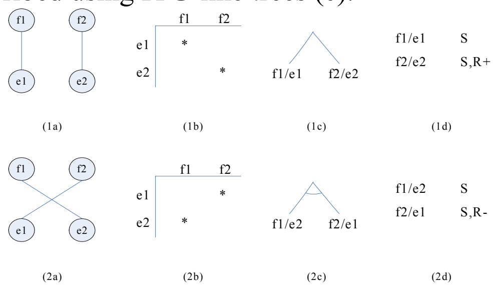


Figure 2: Two basic representations for tree items

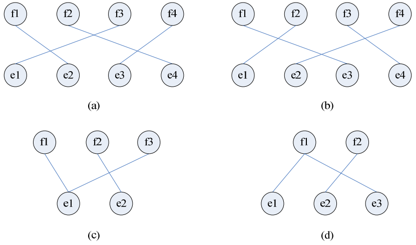


Figure 3: “inside-out” transpositions (a) and (b) with two typical complex sequences (c) and (d). In (c) and (d), word correspondence f2-e2 is also extracted as sub-alignments.

The two widely known situations that cannot be described by ITGs are called “inside-out” transpositions (Figure 3 a & b). Since they cannot be decomposed in ITGs, we consider them as basic units. In this case, phrase alignment is used. In our model, more complex situations exist for the word correspondences are generated automatically from GIZA++. At the same time, we also keep the sub-alignments in those complex situations in order to extend the coverage of translation options. The sub-alignments are restricted to those that can be described by the two basic operations. In other words, for our ITG-like tree, the nodes are mostly word pairs, except some indecomposable word sequences pairs. Figure 3 shows four typical complex sequences viewed as phrases.

Therefore, our ITG-like trees take some phrase alignments into consideration and we also keep the sub-alignments in these situations. Tree items in our model are restricted to minimum constituents for the simplicity of parsing tree generation. Then we extract those word pairs from tree items, instead of all the possible word sequences, as our translation table. In this way, we can greatly reduce the number of translation pairs to be consideration.

3.2 SHIFT and REDUCE Operations

Sagae and Lavie (2005) propose a constituency-based parsing method to determine sentence dependency structures. This method is simple and efficient, which makes use of SHIFT and REDUCE operations within a stack framework. This kind of representations can be easily learned by a classifier with linear time complexity.

In their method, they build a parse tree of a sentence one word at a time just as in a stack parser. At any time step, they either shift a new word on to the stack, or reduce the top two elements on the stack into a new non-terminal.

Sagae and Lavie's algorithms are designed for monolingual parsing problem. We extend it to represent our ITG-like tree. In our problem, each word pairs can be viewed as tree **items** (nodes). To handle our tree alignment problem, we need to define two REDUCE operations: REDUCE in order and REDUCE in reverse order. We define these three basic operations as follows:

- S: SHIFT - push the current item onto the stack.
- R+: REDUCE in order - pop the first two items from the stack, and combine them in the **original order** on the target side, then push back.
- R-: REDUCE in reverse order - pop the first two items from the stack, and combine them in the **reverse order** on the target side, then push back.

Using these operators, our ITG-like tree is transformed to serial stack operations. In Figure 2, (d) is such a representation for the two basic alignments. Therefore, the structure of word aligned sentences can be transformed to an operation sequence, which represents the bilingual parsing correspondences.

After that, we attach these operations to each corresponding tree item like a sequence labeling problem. We need to perform another "grouping" step to make sure only one operation is assigned to each item, such as "S,R+", "S,R-,R+", etc. Then, those grouped operations are regarded as a whole and performed as one label. The number of this kind of labels is decided by the training corpus[1]. Having defined such labels, the prediction of tree structures is transformed to a label prediction one. That is, giving word pairs as input, we transform them to their corresponding labels (stack operations) in the output. At the same time, tree transductions are encoded in those labels. Once all the "labels" are performed, there should be only one element in the stack, i.e. the generating sentence translation pairs. See Appendix A for a more complete example in Chinese-English with our defined operations.

Another constraint we impose is to keep the least number of elements in stack at any time. If two elements on the top of the stack can be combined, we combine them to form a single item. This constraint can avoid having too many possible operations for the last word pair, which may make future predictions difficult.

4 Structured Prediction

SEARN is a machine learning method proposed recently by Daumé III et al. (2007) to solve structured prediction problems. It can produce a high prediction performance without compromising speed, simplicity and generality. By incorporating the search and learning process, SEARN can solve the complex problems without having to perform explicit decoding any more.

In most cases, a prediction of input x in domain X into output y in domain Y, like SVM and decision trees, cannot keep the structure information during prediction. SEARN considers this problem as a cost sensitive classification one. By defining features and a loss function, it performs a cost sensitive learning algorithm to learn predictions. During each iteration, the optimal policy (decided by previous classifiers) generates new training examples through the search space. These data are used to adjust performance for next classifier. Then, iterations can keep this algorithm to perform better for prediction tasks. Structures are preserved for it integrates searching and learning at the same time.

4.1 Parsing Tree Prediction

For our problem, using SEARN to predict the stack-based ITG-like trees, given word alignments as input, can benefit from the advantages of this algorithm. With the structured learning method, we can account for the sentence structures and their correspondence between two languages at

[1] This set of labels is quite small and only 16 for the French-English training set with 688,031 sentences.

the same time. Moreover, it keeps the translating structures from source to target.

As we have transformed the tree-to-tree translation problem into a sequence labeling one, all we need to solve is a tagging problem similar to a POS tagging (Daumé III et al. 2006). The input sequence x is word pairs and output y is the group of SHIFT and REDUCE operations. For sequence labeling problem, the standard loss function is Hamming distance, which measures the difference between the true output and the predicting one:

$$HL(y, \hat{y}) = \sum_t \delta(y_t, \hat{y}_t) \qquad (1)$$

where δ is 0 if two variables are equal, and 1 otherwise.

5 Decoder

We use a left-to-right beam search decoder to find the best translation given a source sentence. Compared with general phrase-based beam search decoder like Pharaoh (Koehn 2004), this decoder integrates structured information and does not need distortion cost and other costs (e.g. future costs) any more. Therefore, the best translation can be determined by:

$$e^* = \arg\max_e \{p(f \mid e) p_{lm}(e) \omega^{length(e)}\} \qquad (2)$$

where ω is a factor of word length penalty. Similarly, the translation probability $p(f \mid e)$ can be further decomposed into:

$$p(f \mid e) = \prod_i \phi(f_i \mid e_i) \qquad (3)$$

and $\phi(f_i \mid e_i)$ represents the probability distribution of word pairs.

Instead of extracting all possible phrases from word alignments, we consider those translation pairs from the nodes of ITG-like trees only. Like Pharaoh, we calculate their probability as a combination of 5 constituents: phrase translation probability (in both directions), lexical translation probability (in both directions) and phrase penalty (default is set at 2.718). The corresponding weight is trained through minimum error rate method (Och 2003). Parameters of this part can be calculated in advance once tree structures are generated and can be stored as phrase translation table.

5.1 Core Algorithm

Another important question is how to preserve sentence structures during decoding. A left-to-right monotonous search procedure is needed. Giving the source sentence, word translation candidates can be determined according to the translation table. Then, several rich features like current and previous source words are extracted based on these translation pairs and source sentence. After that, our structured prediction learning method will be used to predict the output "labels", which produces a bilingual parsing tree. Then, a target output will be generated for the current partial source sentence as soon as bilingual parsing trees are formed. The output of this part therefore contains syntactic information for structure.

For instance, given the current source partial like "f1 f2", we can generate their translation word pair sequences with the translation table, like "f1/e1 f2/e2", "f1/e3 f2/e4" and so on. The corresponding features are then able to be decided for the next predicting process. Once the output predictions (i.e. stack operations) are decided, the bilingual tree structures are formed at the same time. As a consequence, results of these operations are the final translations which we really need.

At each stage of translation, language model parameters can be added to adjust the total costs of translation candidates and make the pruning process reasonable. The whole sentence is then processed by incrementally constructing the translation hypotheses. Lastly, the element in the last beam with the minimum cost is the final translation. In general, the translation process can be described in the following way:

1. Generating all the translation options based on the phrase translation table
2. For each source word in sequence, choose one translation counterpart, generate its word pairs and current word pair sequence for sentence
3. Extract features for structured prediction
4. Using SEARN to predict the "label" sequence and then its parsing tree
5. Generate current target output with recombining and pruning
6. Add the next source word and repeat step 2 until all is finished
7. Output the one with the minimum cost in the last beam.

5.2 Recombining and Pruning

Different translation options can combine to form the same fragment by beam search decoder. Recombining is therefore needed here to reduce the search space. So, only the one with the lowest cost is kept when several fragments are identical. This recombination is a risk-free operation to improve searching efficiency.

Another pruning method used in our system is histogram pruning. Only n-best translations are allowed for the same source part in each stack (e.g. n=100). In contrast with traditional beam search decoder, we generate our translation candidates from the same input, instead of all allowed word pairs elsewhere. Therefore the pruning is much more reasonable for each beam. There is no relative threshold cut off compared with Pharaoh.

In the end, the complexities for decoding are the main concern of our method. In practice, however, it will not exceed the $O(m*N*Tn)$ (m for sentence length, N for stack size and Tn for allowed translation candidates). This is based on the assumption that our prediction process (tackled by SEARN) is fed with three features (only one former item is associated), which makes it no need of full sentence predictions at each time.

6 Experiment

We validate our method using the corpus from the shared task on NAACL 2006 workshop for statistical machine translation[2]. The difference of our method lies in the framework and different phrase translation table. Experiments are carried on all the three language pairs (French-English, German-English and Spanish-English) and performances are evaluated by the providing test sets. System parameters are adjusted with development data under minimum error rate training.

For SEARN, three features are chosen to use: the current source word, the word before it and the current target word. As we do not know the real target word order before decoding, the corresponding target word's position cannot be used as features. Besides, we filter the features less than 5 times to reduce the training complexities.

The classifier we used in the training process is based on perceptron because of its simplicity and performance. We modified Daumé III's script[3] to fit our method and use the default 5 iterations for each perceptron-based training and 3 itertaions for SEARN.

6.1 Results for different language pairs

The final results of our system, named **Amasis**, and baseline system Pharaoh (Koehn and Monz 2006) for three language pairs are listed in Table 1. The last three lines are the results of Pharaoh with phrase length from 1 to 3. However, the length of phrases for Amasis is determined by ITG-like tree nodes and there is no restriction for it.

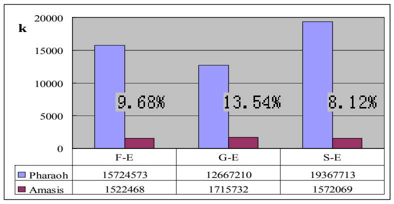


Figure 4: Numbers of translation table

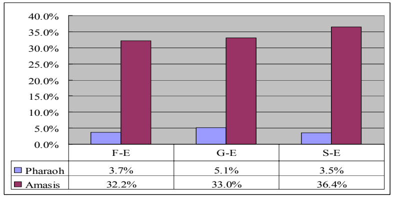


Figure 5: Percent of single word translation pairs (only one word in the source side)

	F-E		G-E		S-E	
	In	Out	In	Out	In	Out
Amasis	27.44	18.41	23.02	15.97	27.51	23.35
Pharaoh$_1$	20.54	14.07	17.53	12.13	23.23	20.24
Pharaoh$_2$	27.71	19.41	23.36	15.77	28.88	25.28
Pharaoh$_3$	30.01	20.77	24.40	16.58	30.58	26.51

Table 1: BLEU scores for different language pairs. In - In-domain test, Out - Out-of-domain test.

Even without producing higher BLEU scores than Pharaoh, our approach is still interesting for the following reasons. First, the number of phrase translation pairs is greatly reduced in our system. The ratio of translation table number in our method (Amasis) to Pharaoh, for French-English is 9.68%, for German-English is 13.54%, for Spanish-English is 8.12% (Figure 4). This means that our method is more efficient at combining words and phrases during translation. The reasons for the different ratio for the three languages are not very clear, maybe are related to the flexibility of word order of source language. Second, we count the single word translation pairs (only one word in the source side) as shown in Figure 5. There are significantly more single word translations in our method. However, the translation quality can be kept at the same level under this circumstance. Third, our current experimental results are produced with only three common features (the corresponding current source and target word and the last source one) without any linguistics information. More useful features are expected to be helpful like POS tags. Finally, the performance can be further improved if we use a more powerful classifier (such as SVM or ME) with more iterations.

[2] http://www.statmt.org/wmt06/shared-task/

[3] http://www.cs.utah.edu/~hal/searn/SimpleSearn.tgz

7 Conclusion

Our method provides a simple and efficient way to solve the word ordering problem partially which is NP-hard (Knight 1999). It is word based except for those indecomposable word sequences under ITGs. However, it can achieve comparable results with phrase-based method (like Pharaoh), while much fewer translation options are used. For the structure prediction process, only 3 common features are preserved and perceptron-based classifiers are chosen for the use of simplicity. We argue that this approach is promising when more features and more powerful classifiers are used as Daumé III et al. (2007) stated.

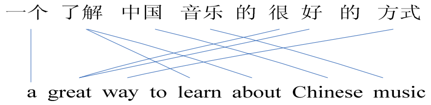
Our contributions lie in the integration of structure prediction for bilingual parsing trees through serial transformations. We reinforce the power of formally syntax-based method by using structured prediction method to obtain tree-to-tree transductions by the transforming from word alignments to ITG-like trees and then to label sequences. Thus, the sentence structures can be better accounted for during translating.

Acknowledgements

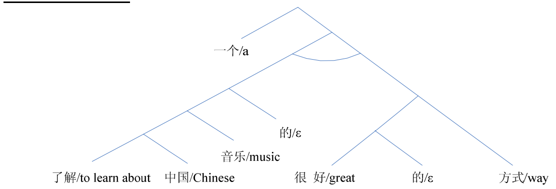
This work is partially supported by National Natural Science Foundation of China under grant #60773027, #60736044 and by "863" Key Projects #2006AA010108. We would like to thank anonymous reviewers for their detailed comments.

Appendix A. A Complete Example in Chinese-English with Our Defined Operations

Word alignments


一个 了解 中国 音乐 的 很 好 的 方式
a great way to learn about Chinese music

ITG-like tree



SHIFT-REDUCE label sequence

一个/a	S
了解/to learn about	S
中国/Chinese	S,R+
音乐/music	S,R+
的/ε	S,R+
很 好/great	S
的/ε	S,R+
方式/way	S,R+,R-,R+

Stack status when operations finish

一个 了解 中国 音乐 的 很 好 的 方式
/ a great way to learn about Chinese music

References

A. Burbank, M. Carpuat, et al. 2005. Final Report of the 2005 Language Engineering Workshop on Statistical Machine Translation by Parsing. Johns Hopkins University

D. Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In ACL, pages 263-270.

M. Galley, M. Hopkins, et al. 2004. What's in a translation rule? In HLT-NAACL, Boston, MA.

D. Gildea. 2003. Loosely Tree-Based Alignment for Machine Translation. In ACL, pages 80-87, Sapporo, Japan.

H. Daumé III, J. Langford, et al. 2007. Search-based Structured Prediction. Under review by the Machine Learning Journal. http://pub.hal3.name/daume06searn.pdf.

H. Daumé III, J. Langford, et al. 2006. Searn in Practice. http://pub.hal3.name/daume06searn-practice.pdf.

K. Knight. 1999. Decoding Complexity in Word-Replacement Translation Models. Computational Linguistics 25(4): 607-615.

K. Knight and J. Graehl. 2005. An Overview of Probabilistic Tree Transducers for Natural Language Processing. In CICLing, pages 1-24.

P. Koehn. 2004. Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In Proc. of AMTA, pages 115-124.

P. Koehn and C. Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In Proc. on the Workshop on Statistical Machine Translation, pages 102-121, New York City.

P. Koehn, F. J. Och, et al. 2003. Statistical Phrase-Based Translation. In HLT-NAACL, pages 127-133.

P. Liang, A. Bouchard, et al. 2006. An End-to-End Discriminative Approach to Machine Translation. In ACL.

Y. Liu, Q. Liu, et al. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In ACL.

F. J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In ACL, pages 160-167.

F. J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics 29(1): 19-51.

K. Sagae and A. Lavie. 2005. A Classifier-Based Parser with Linear Run-Time Complexity. In IWPT, pages 125-132.

C. Tillmann and T. Zhang. 2005. A Localized Prediction Model for Statistical Machine Translation. In ACL.

C. Tillmann and T. Zhang. 2006. A Discriminative Global Training Algorithm for Statistical MT. in ACL.

J. Turian, B. Wellington, et al. 2006. Scalable Discriminative Learning for Natural Language Parsing and Translation. In Proceedings of NIPS, Vancouver, BC.

T. Watanabe, J. Suzuki, et al. 2007. Online Large-Margin Training for Statistical Machine Translation. In EMNLP.

D. Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics 23(3): 377-404.

D. Xiong, Q. Liu, et al. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In ACL, pages 521-528.

K. Yamada and K. Knight. 2001. A Syntax-based Statistical Translation Model. In ACL, pages 523-530.

K. Yamada and K. Knight. 2002. A Decoder for Syntax-based Statistical MT. In ACL, pages 303-310.