Improving the Hierarchical Phrase-Based Translation Model

Xiaodong Shi, Xiang Zhu, Yidong Chen
Institute of Artificial Intelligence
Xiamen University
Xiamen, China 361005
{mandel,ydchen}@xmu.edu.cn

Abstract

The classic Hierarchical Phrase-Based Translation Model suffers from 3 defects: 1) an overly large model with a lot of useless or even wrong translation rules; 2) a large search space leading to combinatorial explosion; 3) since there is only one variable, decoder can only choose hierarchical rules through lexical information. This paper presents strategies to mitigate these problems, including rule extraction filtering, decoding optimization and variable refinement. The experiments show that the proposed methods can not only speed the decoding process but also improve the translation quality.

1 Introduction

Since its inception, David Chiang's (Chiang, 2005; Chiang, 2007) has met with considerable success and is one of the best models for Statistical Machine Translation. However, it also has a number of obvious disadvantages:

1) Since Chiang's hierarchical rule[*] extraction method is relatively simple, it tends to produce a large number of useless and unreliable rules, which cause many problems, e.g. an overly large model, slow decoding, and decoding errors. To mitigate these problems, we introduce a rule extraction filtering strategy, in which we measure the generalization ability of each rule and filter out rules that prove inadequate in this regard.

2) During decoding, even with Cube pruning, CKY algorithm searches too many decoding paths which results in excessive time consumption and numerous decoding errors. To mitigate this problem, we introduce a decoding optimization strategy, which prevent the decoder from some unreasonable decoding paths.

3) Because there is only one variable in the Hierarchical Phrase-based model, during decoding, the decoder has no guide to choose appropriate hierarchical rules except for relying on the lexical information and language model, a limitation which causes further decoding errors. To mitigate this problem, we propose a framework of variable refinement.

The remainder of this paper is organized as follows: Section 2 introduces the rule extraction filtering strategy. Section 3 describes the decoding optimization strategy. Section 4 presents the framework of variable refinement and a preliminary variable refinement scheme based on part-of-speech tags within the framework. In each section we have run experiments to show the effectiveness of our methods. We conclude in Section 5.

2 A Rule Extraction Filtering Strategy Based on the Generalization Ability

In the Hierarchical Phrase-Based Translation Model many extracted translation rules are wrong or redundant, which cause slow and erroneous decoding. We propose to measure the usefulness of a hierarchical rule through its generalization ability and filter out rules that cannot generalize because these can be replaced by initial rules and glue rules. We have found that these rules are often obtained from wrong initial rules, which are in turn formed from bad alignment. Filtering not only reduces model size and also result in less errors.

[*] We use rules or phrases interchangeably.

2.1 Rule Extraction Filtering

In general, we think that a general syntactic pattern will appear frequently in a corpus big enough, so a hierarchical rule (called H-rule henceforth) which can generalize will be extracted more than once from different initial rules. Our method is very simple: if an H-rule is extracted only once, the rule should be discarded. We illustrate our method by an example. Suppose an H-rule

"I am having [X]→ Wo zhenzai ci [X]" (1)

is extracted from the initial rule

"I am having breakfast→ Wo zhenzai ci fan"

By replacing the initial rule

"breakfast→ fan" (2)

with variables, we add the rule (2) to the Base Rule Set (BRS) of the H-rule (1). If finally the cardinality of the BRS of an H-rule is 1, then the H-rule is discarded. We then obtain a new H-rule set different from the classic Hierarchical model.

2.2 Experiments and Analysis

We tested our filtering strategy in an English-Chinese direction. We use a corpus with 396475 pairs of aligned sentences, in which the English is 44.7MB and the Chinese is 42.5MB. We use a Chinese trigram language model of 1.69GB. We use two test sets from China Workshop on Machine Translation 2008 (Hongmei Zhao et al., 2008). The following tables show the data sets and results. We call our model the rule filtering model. We use BLEU-SBP (Chiang et al., 2008) as our translation quality measurement.

Table 2.1 Model sizes

Model	Initial rules	Rules With One Variables	Rules With Two Variables
Chang's Model	5388313	15967381	8285387
Our Model	5388313	3730414	909000

Table 2.2 Test set sizes

Test sets	Size
Test set 1	142KB/995 sentences
Test set 2	134KB/1008 sentences

Table 2.3 Results for test set 1

Model	Decoding Time (in seconds)	BLEU-SBP
Chang's Model	3416	0.2548
Our Model	2699	0.2618

Table 2.4 Results for test set 2

Model	Decoding Time (in seconds)	BLEU-SBP
Chang's Model	3298	0.3775
Our Model	2594	0.3839

As we can see from the above tables, our model discarded 75% single variable H-rules and nearly 90% two-variable H-rules, which then results in faster decoding, and the translation quality is even better than Chiang's original model.

3 Decoding Optimization

3.1 Motivation

Proper use of proper rules will result in good translation, but not all rules are good. Let's illustrate this by an example. To translate

"I ate a pear in the morning" (3)

We can use the following proper rules:

"a pear -> yi ge li" (4)

"I ate [X] -> wo ci le [X]" (5)

"[X] in the morning -> zaoshang [X]" (6)

And get the correct translation, as in Figure 1.

However, we have also the following rules in our model:

"pear in the -> zhong de li" (7)

"I ate a [X] -> wo ci le yi ge [X]" (8)

" morning -> zaoshang " (9)

Using these rules we might get the incorrect translation in Figure 2.

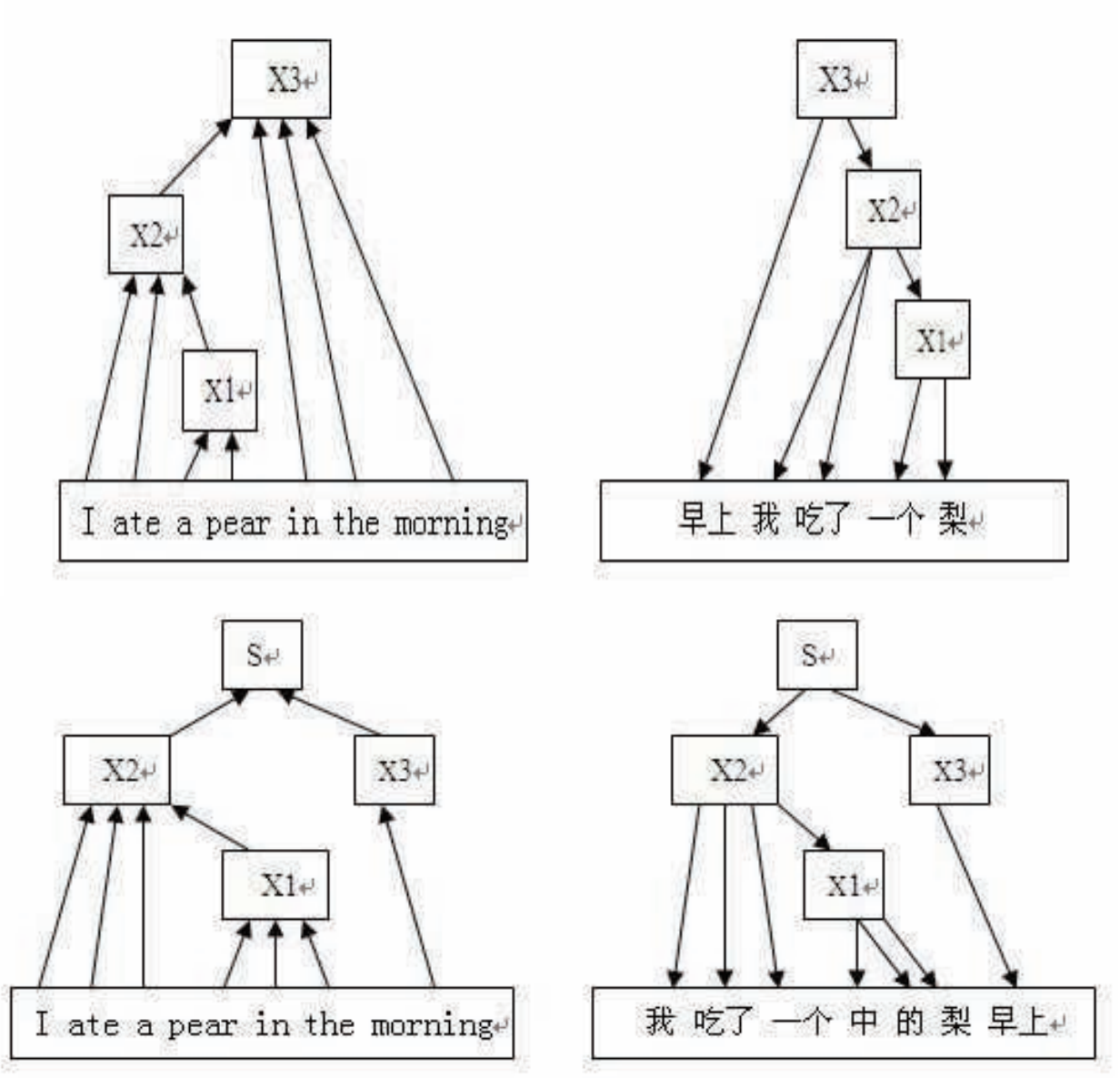
Although rule (7) cannot be said to be wrong from word alignment, it should not be used in the decoding of this sentence. We notice that "a pear" is a proper translation unit, while "pear in the" is not. We need to ban the latter from decoding.

We define a phrase to be *improper* if it's composed of an incomplete noun phrase fragment and one or more surrounding words. In the sentence (3) above, phases "ate a", "pear in", "pear in the" are improper. We choose noun phrase because it's

used very often and its proper identification has a major impact on the quality of translation.

In decoding, we first analyze the source sentence using the Berkeley parser (Petrov and Klein, 2007) and then collect all improper phrases into a set called Y, and we modify the search algorithm as described in Chiang (2005) and Chiang (2007) as follows:

```
procedure PARSE
    for all axioms (X→γ) do
      if γ∉ Y then add(X→γ) to rchart
    for ℓ ←1...n do
      for all i,j s.t. j-i=ℓ do
        if ℓ ≦ Λ then
          for all items [X,i,j]: ω inferable from items
              in rchart and chart do
            if γ∉ Y then add [X,i,j]: ω to chart[X,i,j]
        if i=0 then
          for all items [S,i,j] : ω inferable from
              items in rchart and chart do
            if γ∉ Y then add [S,i,j] : ω to chart [S,i,j]
```



3.2 Experiments and Analysis

We use the same training and test sets as described in Section 2. We use two models as benchmarks: Baseline 1 is Chiang's original model and Baseline 2 is our filtering model (abbreviated as BL1, and

BL2 in Tables 3.1 and 3.2). We do search optimization on both models and call them BL1++ and BL2++. We also given the parsing time of the test set.

Table 3.1 Parsing time

Test set	Parsing time (seconds)
Test set 1	809
Test set 2	595

Table 3.2 Results for test set 1

Model	Decoding Time (in seconds)	BLEU-SBP
BL1	3416	0. 2548
BL1++	1245	0. 2582
BL2	2699	0. 2618
BL2++	1078	0. 2649

Table 3.2 Results for test set 2

Model	Decoding Time (in seconds)	BLEU-SBP
BL1	3298	0. 3775
BL1++	1367	0. 3794
BL2	2594	0. 3839
BL2++	1179	0. 3866

We can see from the above tables that even with added time of parsing the decoding is still faster than benchmarks and the translation quality are better.

4 Variable Refinement

4.1 Framework

In an H-rule, a variable can be replaced by any phrase type. This unfortunately will lead to many errors. For example, "I had [X]" can mean "I ate [X]", only for certain types of objects, e.g. "dinner". To avoid translate "I had a friend" into something like "I ate a friend", we need to replace [X] by [X-meal] or the like.

When we extract an H-rule from an initial rule by replacing the initial rule with an [X], we should use a more specific [X], e.g. [X1] if the type of the initial rule can be identified as [X1]. We can introduce any number of variable types if necessary as long as the types can be identified reliably. Of course an initial rule can also belong to many types.

To avoid the loss of translation ability, we need not discard the original untyped H-rule. The idea is to use the more specific type in decoding if applicable. Although more H-rules are generated now, the decoding is actually faster, because if the type of the initial phrase can be identified, we only use more specific rules; otherwise we use the original more general rule.

Variable refinement is not new, e.g. Libin Shen et al (2009) explored the idea of labeling nonterminals in the target side with the POS tag of its headword in a string-to-dependency model. Our approach differs from theirs in that our variable refinement operates on the source side and has no structure restrictions otherwise.

4.2 Variable Refinement by Part of Speech

Part of Speech (POS) is very important for the meaning of a word (Brill 1995; Jurafsky and Martin, 2008), e.g. *book* as a noun is very different from *book* as a verb. An H-rule extracted from an initial phase with a POS type cannot usually be applied to an initial phrase with a different POS type. We thus define a variable refinement scheme based on POS as follows:

1) Pos-tag the source side of the training sentences;

2) Put those initial phrases with only one source word and with its POS belonging to one of noun, verb, adjective, adverb, propositions into five categories: N, V, ADJ, ADV, P; all other initial phrases belong to category X;

3) Extract H-rules with the same type and X type from initial rules categories N, V, ADJ, ADV and P;

4) H-rules extracted from Category X apply to all initial rules, and H-rules of other type apply to the specific initial rules;

5) Add glue rules to allow for gluing of phrases of any types;

6) In decoding, we pos-tag the source sentence to identify nouns, verbs, adjective, adverbs and propositions and tag them as N, V, ADJ, ADV, P respectively and these words can only use their specific H-rules; other phrases can use untyped H-rules.

Thus, we have a preliminary scheme of variable refinement based on the POS tags.

4.3 Experiments and Analysis

The variable refinement model added many H-rules for some initial rules, resulting in a much bigger model. So we trained on a small data set. We use the 36170 parallel English-Chinese sentences of the Penn Treebank as the training corpus, with an English of 5.95MB and Chinese of 5.20MB. The trained model sizes are described in the following table (in which BL2 is the filtering model of Section 2 and BL2+v adds variable refinement):

Table 4.1 Model sizes

Model	Initial rules	Rules With One Variables	Rules With Two Variables
Chang's Model	434822	870748	377740
BL2	434822	224298	39451
BL2+v	434822	1345788	1420236

We need to pos-tag the test set before decoding, so we give the pos-tagging time in the following table:

Table 4.2 Pos-tagging time

Test set	time (seconds)
Test set 1	12.9
Test set 2	12.7

The results are shown in Tables 4.3 and 4.4:

Table 4.3 Results for test set 1

Model	Decoding Time (in seconds)	BLEU-SBP
Chang's Model	3026	0. 2356
BL2	2622	0. 2391
BL2+v	1750	0. 2448

Table 4.4 Results for test set 2

Model	Decoding Time (in seconds)	BLEU-SBP
Chang's Model	2508	0. 1671
BL2	2101	0. 1682
BL2+v	1411	0. 1727

We can see from the above tables that even with much bigger model size and added time of pos-tagging, the decoding is still faster than benchmarks and the translation quality are better than Chiang's original model.

5 Conclusions

We discussed the inadequacies of the Hierarchical Phrase model and described three ways to improve the models: rule extraction filtering based on generalization ability, search optimization on decoding and variable refinement. Experiments show that both decoding efficiency and translation quality are improved.

More work need to be done on variable refinement and we think this is one of the very promising direction to pursue to improve the statistical machine translation model.

Acknowledgments

The work described in this paper is partially sponsored by Fujian Provincial science foundation grant No. 2006J0043, No. 2006H0038 and national high-tech grant No. 2006AA010108.

Thanks for the one of the reviewers' comments on variable refinement.

References

Daniel Jurafsky, James H. Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (2ed.)*, Prentice Hall.

David Chiang. 2005. *A Hierarchical Phrase-Based Model for Statistical Machine Translation*. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.

David Chiang. 2007. *Hierarchical Phrase-Based Translation*. Association for Computational Linguistic.

David Chiang, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. *Decomposability of translation metrics for improved evaluation and efficient algorithms*. In Proc. EMNLP 2008, pages 610-619.

Eric Brill. 1995. *Transformation-based Error-driven Learning and Natural Language Processing : A Case Study in Part-of-speech Tagging*. Computational Linguistics ,1995, 21(4):543-565.

Hongmei Zhao, Jun Xie, Qun Liu, Yajuan Lü Dongdong Zhang, Mu Li. 2008. *Introduction to China's CWMT2008 Machine Translation Evaluation.* Procedings of CWMT 2008.

Libin Shen, Bing Zhang, Spyros Matsoukas, Jinxi Xu and Ralph Weischedel. 2010. *Statistical Machine Translation with a Factorized Grammar.* In Proceedings of Conference on Empirical Methods in Natural Language Processing. Cambridge, MA USA, Oct. 9 - 11, 2010.

Slav Petrov and Dan Klein, 2007. *Improved Inference for Unlexicalized Parsing*. In Proceeding of HLT-NAACL 2007