What Can Syntax-based MT Learn from Phrase-based MT?

Steve DeNeefe and Kevin Knight
Information Sciences Institute
The Viterbi School of Engineering
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{sdeneefe,knight}@isi.edu

Wei Wang and Daniel Marcu
Language Weaver, Inc.
4640 Admiralty Way, Suite 1210
Marina del Rey, CA 90292
{wwang,dmarcu}@languageweaver.com

Abstract

We compare and contrast the strengths and weaknesses of a syntax-based machine translation model with a phrase-based machine translation model on several levels. We briefly describe each model, highlighting points where they differ. We include a quantitative comparison of the phrase pairs that each model has to work with, as well as the reasons why some phrase pairs are not learned by the syntax-based model. We then evaluate proposed improvements to the syntax-based extraction techniques in light of phrase pairs captured. We also compare the translation accuracy for all variations.

1 Introduction

String models are popular in statistical machine translation. Approaches include word substitution systems (Brown et al., 1993), phrase substitution systems (Koehn et al., 2003; Och and Ney, 2004), and synchronous context-free grammar systems (Wu and Wong, 1998; Chiang, 2005), all of which train on string pairs and seek to establish connections between source and target strings. By contrast, explicit syntax approaches seek to directly model the relations learned from parsed data, including models between source trees and target trees (Gildea, 2003; Eisner, 2003; Melamed, 2004; Cowan et al., 2006), source trees and target strings (Quirk et al., 2005; Huang et al., 2006), or source strings and target trees (Yamada and Knight, 2001; Galley et al., 2004).

It is unclear which of these important pursuits will best explain human translation data, as each has advantages and disadvantages. A strength of phrase models is that they can acquire all phrase pairs consistent with computed word alignments, snap those phrases together easily by concatenation, and re-order them under several cost models. An advantage of syntax-based models is that outputs tend to be syntactically well-formed, with re-ordering influenced by syntactic context and function words introduced to serve specific syntactic purposes.

A great number of MT models have been recently proposed, and other papers have gone over the expressive advantages of syntax-based approaches. But it is rare to see an in-depth, quantitative study of strengths and weaknesses of particular models with respect to each other. This is important for a scientific understanding of how these models work in practice. Our main novel contribution is a comparison of phrase-based and syntax-based extraction methods and phrase pair coverage. We also add to the literature a new method of improving that coverage. Additionally, we do a careful study of several syntax-based extraction techniques, testing whether (and how much) they affect phrase pair coverage, and whether (and how much) they affect end-to-end MT accuracy. The MT accuracy tests are needed because we want to see the individual effects of particular techniques under the same testing conditions. For this comparison, we choose a previously established statistical phrase-based model (Och and Ney, 2004) and a previously established statistical string-to-tree model (Galley et al., 2004). These two models are chosen because they are the basis of two of the most successful systems in the NIST 2006 MT

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 755–763, Prague, June 2007.

evaluation[1].

2 Phrase-based Extraction

The Alignment Template system (ATS) described by Och and Ney (2004) is representative of statistical phrase-based models. The basic unit of translation is the phrase pair, which consists of a sequence of words in the source language, a sequence of words in the target language, and a vector of feature values which describe this pair's likelihood. Decoding produces a string in the target language, in order, from beginning to end. During decoding, features from each phrase pair are combined with other features (e.g., re-ordering, language models) using a log-linear model to compute the score of the entire translation.

The ATS phrase extraction algorithm learns these phrase pairs from an aligned, parallel corpus. This corpus is conceptually a list of tuples of <source sentence, target sentence, bi-directional word alignments> which serve as training examples, one of which is shown in Figure 1.

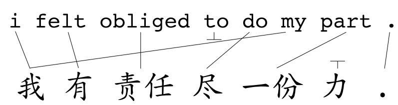

Figure 1: a phrase-based training example

For each training example, the algorithm identifies and extracts all pairs of <source sequence, target sequence> that are consistent with the alignments. It does this by first enumerating all source-side word sequences up to a length limit L, and for each source sequence, it identifies all target words aligned to those source words. For example, in Figure 1, for the source phrase 有 责任 尽, the target words it aligns to are `felt`, `obliged`, and `do`. These words, and all those between them, are the proposed target phrase. If no words in the proposed target phrase align to words outside of the source phrase, then this phrase pair is extracted.

The extraction algorithm can also look to the left and right of the proposed target phrase for neighboring unaligned words and extracts phrases. For example, for the phrase pair 有 责任 ↔ `felt obliged`, the word `to` is a neighboring unaligned word. It constructs new target phrases by adding on consecutive unaligned words in both directions, and extracts those in new pairs, too (e.g., 有 责任 ↔ `felt obliged to`). For efficiency reasons, implementations often skip this step.

Figure 2 shows the complete set of phrase pairs up to length 4 that are extracted from the Figure 1 training example. Notice that no extracted phrase pair contains the character 我. Because of the alignments, the smallest legal phrase pair, 我 有 责任 尽 ↔ `i felt obliged to do my`, is beyond the size limit of 4, so it is not extracted in this example.

有	↔	felt
有 责任	↔	felt obliged
有 责任 尽	↔	felt obliged to do
责任	↔	obliged
责任 尽	↔	obliged to do
尽	↔	do
一份	↔	part
一份 力	↔	part
一份 力 .	↔	part .
力 .	↔	.
.	↔	.

Figure 2: phrases up to length 4 extracted from the example in Figure 1

Phrase pairs are extracted over the entire training corpus. Due to differing alignments, some phrase pairs that cannot be learned from one example may be learned from another. These pairs are then counted, once for each time they are seen in a training example, and these counts are used as the basis for maximum likelihood probability features, such as $p(f|e)$ and $p(e|f)$.

3 Syntax-based Extraction

The GHKM syntax-based extraction method for learning statistical syntax-based translation rules, presented first in (Galley et al., 2004) and expanded on in (Galley et al., 2006), is similar to phrase-based extraction in that it extracts rules consistent with given word alignments. A primary difference is the use of syntax trees on the target side, rather than sequences of words. The basic unit of translation is the translation rule, consisting of a sequence of words

[1] http://www.nist.gov/speech/tests/mt/mt06eval_official_results.html

and variables in the source language, a syntax tree in the target language having words or variables at the leaves, and again a vector of feature values which describe this pair's likelihood. Translation rules can:

- look like phrase pairs with syntax decoration:

  NPB(NNP(prime)
  NNP(minister)
  NNP(keizo)
  NNP(obuchi)) $\leftrightarrow$ 小渊惠三 首相

- carry extra contextual constraints:

  VP(VBD(said)
  x_0:SBAR-C) $\leftrightarrow$ 说 x_0

  (according to this rule, 说 can translate to said only if some Chinese sequence to the right of 说 is translated into an SBAR-C)

- be non-constituent phrases:

  VP(VBD(said)
  SBAR-C(IN(that)
  x_0:S-C)) $\leftrightarrow$ 说 x_0

  VP(VBD(pointed)
  PRT(RP(out))
  x_0:SBAR-C) $\leftrightarrow$ 指出 x_0

- contain non-contiguous phrases, effectively "phrases with holes":

  PP(IN(on)
  NP-C(NPB(DT(the)
  x_0:NNP))
  NN(issue)))) $\leftrightarrow$ 在 x_0 问题 上

  PP(IN(on)
  NP-C(NPB(DT(the)
  NN(issue))
  x_0:PP)) $\leftrightarrow$ 在 x_0 问题 上

- be purely structural (no words):

  S(x_0:NP-C x_1:VP) $\leftrightarrow$ x_0 x_1

- re-order their children:

  NP-C(NPB(DT(the)
  x_0:NN)
  PP(IN(of)
  x_1:NP-C)) $\leftrightarrow$ x_1 的 x_0

Decoding with this model produces a tree in the target language, bottom-up, by parsing the foreign string using a CYK parser and a binarized rule set (Zhang et al., 2006). During decoding, features from each translation rule are combined with a language model using a log-linear model to compute the score of the entire translation.

The GHKM extractor learns translation rules from an aligned parallel corpus where the target side has been parsed. This corpus is conceptually a list of tuples of <source sentence, target tree, bi-directional word alignments> which serve as training examples, one of which is shown in Figure 3.

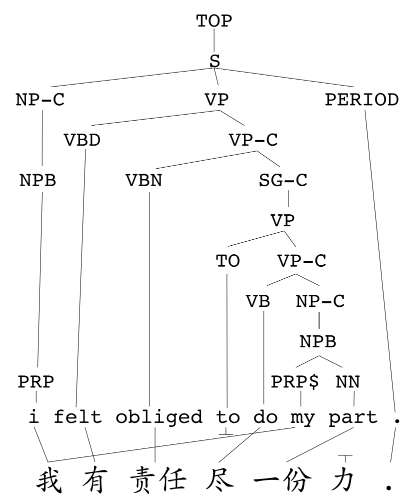

Figure 3: a syntax-based training example

For each training example, the GHKM extractor computes the set of minimally-sized translation rules that can explain the training example while remaining consistent with the alignments. This is, in effect, a non-overlapping tiling of translation rules over the tree-string pair. If there are no unaligned words in the source sentence, this is a unique set. This set, ordered into a tree of rule applications, is called the derivation tree of the training example. Unlike the ATS model, there are no inherent size limits, just the constraint that the rules be as small as possible for the example.

Ignoring the unaligned 力 for the moment, there are seven minimal translation rules that are extracted from the example in Figure 3, as shown in Figure 4. Notice that rule 6 is rather large and applies to a very limited syntactic context. The only constituent node that covers both i and my is the S, so the rule rooted at S is extracted, with variables for every branch below this top constituent that can be explained by other rules. Note also that to becomes a part of this rule naturally. If the alignments were not as constraining (e.g., if `my` was unaligned), then instead of this one big rule many smaller rules would be extracted, such as structural rules (e.g., VP(x_0:VBD x_1:VP-C) $\leftrightarrow x_0\, x_1$) and function word insertion rules (e.g., VP(TO(to) x_0:VP-C) $\leftrightarrow x_0$).

```
1. VBD(felt) ↔ 有
2. VBN(obliged) ↔ 责任
3. VB(do) ↔ 尽
4. NN(part) ↔ 一份
5. PERIOD(.) ↔ .
6. S(NP-C(NPB(PRP(I)))
     VP(x0:VBD
        VP-C(x1:VBN
              SG-C(VP(TO(to)
                      VP-C(x2:VB
                            NP-C(NPB(PRP$(my)
                                      x3:NN)))))))
     x4:PERIOD) ↔ 我 x0 x1 x2 x3 x4
7. TOP(x0:S) ↔ x0
```

Figure 4: rules extracted from training example

We ignored unaligned source words in the example above. Galley et al. (2004) attach the unaligned source word to the highest possible location, in our example, the S. Thus it is extracted along with our large rule 6, changing the target language sequence to "我 x_0 x_1 x_2 x_3 力 x_4". This treatment still results in a unique derivation tree no matter how many unaligned words are present.

In Galley et al. (2006), instead of a unique derivation tree, the extractor computes several derivation trees, each with the unaligned word added to a different rule such that the data is still explained. For example, for the tree-string pair in Figure 3, 力 could be added not only to rule 6, but alternatively to rule 4 or 5, to make the new rules:

NN(part) ↔ 一份 力

PERIOD(.) ↔ 力 .

This results in three different derivations, one with the 力 character in rule 4 (with rules 5 and 6 as originally shown), another with the 力 character in rule 5 (with rules 4 and 6 as originally shown), and lastly one with the 力 character in rule 6 (with rules 4 and 5 as originally shown) as in the original paper (Galley et al., 2004). In total, ten different rules are extracted from this training example.

As with ATS, translation rules are extracted and counted over the entire training corpus, a count of one for each time they appear in a training example. These counts are used to estimate several features, including maximum likelihood probability features for $p(e_{tree}, f_{words}|e_{head})$, $p(e_{words}|f_{words})$, and $p(f_{words}|e_{words})$.

4 Differences in Phrasal Coverage

Both the ATS model and the GHKM model extract linguistic knowledge from parallel corpora, but each has fundamentally different constraints and assumptions. To compare the models empirically, we extracted phrase pairs (for the ATS model) and translation rules (for the GHKM model) from parallel training corpora described in Table 1. The ATS model was limited to phrases of length 10 on the source side, and length 20 on the target side. A superset of the parallel data was word aligned by GIZA union (Och and Ney, 2003) and EMD (Fraser and Marcu, 2006). The English side of training data was parsed using an implementation of Collins' model 2 (Collins, 2003).

	Chinese	Arabic
Document IDs	LDC2003E07	LDC2004T17
	LDC2003E14	LDC2004T18
	LDC2005T06	LDC2005E46
# of segments	329,031	140,511
# of words in foreign corpus	7,520,779	3,147,420
# of words in English corpus	9,864,294	4,067,454

Table 1: parallel corpora used to train both models

Table 2 shows the total number of GHKM rules extracted, and a breakdown of the different kinds of rules. Non-lexical rules are those whose source side is composed entirely of variables — there are no source words in them. Because of this, they potentially apply to any sentence. Lexical rules (their counterpart) far outnumber non-lexical rules. Of the lexical rules, a rule is considered a *phrasal rule* if its source side and the yield of its target side contain exactly one contiguous phrase each, optionally with one or more variables on either side of the phrase. Non-phrasal rules include structural rules, re-ordering rules, and non-contiguous phrases. These rules are not easy to directly compare to any phrase pairs from the ATS model, so we do not focus on them here.

Phrasal rules can be directly compared to ATS phrase pairs, the easiest way being to discard the

Statistic	Chinese	Arabic
total translation rules	2,487,110	662,037
non-lexical rules	110,066	15,812
lexical rules	2,377,044	646,225
phrasal rules	1,069,233	406,020
distinct GHKM-derived phrase pairs	919,234	352,783
distinct corpus-specific GHKM-derived phrase pairs	203,809	75,807

Table 2: a breakdown of how many rules the GHKM extraction algorithm produces, and how many phrase pairs can be derived from them

syntactic context and look at the phrases contained in the rules. The second to last line of Table 2 shows the number of phrase pairs that can be derived from the above phrasal rules. The number of GHKM-derived phrase pairs is lower than the number of phrasal rules because some rules represent the same phrasal translation, but with different syntactic contexts. The last line of Table 2 shows the subset of phrase pairs that contain source phrases found in our development corpus.

Table 3 compares these corpus-specific GHKM-derived phrase pairs with the corpus-specific ATS phrase pairs. Note that the number of phrase pairs derived from the GHKM rules is less than the number of phrase pairs extracted by ATS. Moreover, only slightly over half of the phrase pairs extracted by the ATS model are common to both models. The limits and constraints of each model are responsible for this difference in contiguous phrases learned.

Source of phrase pairs	Chinese	Arabic
GHKM-derived	203,809	75,807
ATS	295,537	133,576
Overlap between models	160,901	75,038
GHKM only	42,908	769
ATS only	134,636	58,538
ATS-useful only	1,994	2,199

Table 3: comparison of corpus-specific phrase pairs from each model

GHKM learns some contiguous phrase pairs that the phrase-based extractor does not. Only a small portion of these are due to the fact that the GHKM model has no inherent size limit, while the phrase based system has limits. More numerous are cases where unaligned English words are not added to an ATS phrase pair while GHKM adopts them at a syntactically motivated location, or where a larger rule contains mostly syntactic structure but happens to have some unaligned words in it. For example, consider Figure 5. Because `basic` and `will` are unaligned, ATS will learn no phrase pairs that translate to these words alone, though they will be learned as a part of larger phrases.

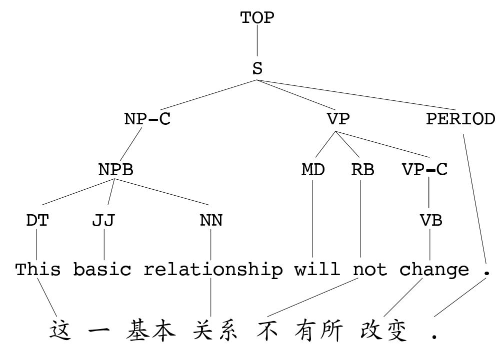

Figure 5: Situation where GHKM is able to learn rules that translate into `basic` and `will`, but ATS is not

GHKM, however, will learn several phrasal rules that translate to `basic`, based on the syntactic context

NPB(x_0:DT
JJ(basic) $\leftrightarrow x_0$ 一 x_1
x_1:NN)

NPB(x_0:DT
JJ(basic) $\leftrightarrow x_0$ 一基本 x_1
x_1:NN)

NPB(x_0:DT
JJ(basic) $\leftrightarrow x_0$ 基本 x_1
x_1:NN)

and one phrasal rule that translates into `will`

VP(MD(will)
x_0:RB $\leftrightarrow x_0$ 有所 x_1
x_1:VP-C)

The quality of such phrases may vary. For example, the first translation of 一 (literally: "one" or "a") to `basic` above is a phrase pair of poor quality, while the other two for `basic` and one for `will` are arguably reasonable.

However, Table 3 shows that ATS was able to learn many more phrase pairs that GHKM was not. Even more significant is the subset of these missing phrase pairs that the ATS decoder used in its best[2]

[2]i.e. highest scoring

translation of the corpus. According to the phrase-based system these are the most "useful" phrase pairs and GHKM could not learn them. Since this is a clear deficiency, we will focus on analyzing these phrase pairs (which we call *ATS-useful*) and the reasons they were not learned.

Table 4 shows a breakdown, categorizing each of these missing ATS-useful phrase pairs and the reasons they were not able to be learned. The most common reason is straightforward: by extracting only the minimally-sized rules, GHKM is unable to learn many larger phrases that ATS learns. If GHKM can make a word-level analysis, it will do that, at the expense of a phrase-level analysis. Galley et al. (2006) propose one solution to this problem and Marcu et al. (2006) propose another, both of which we explore in Sections 5.1 and 5.2.

Category of missing ATS-useful phrase pairs	Chinese	Arabic
Not minimal	1,320	1,366
Extra target words in GHKM rules	220	27
Extra source words in GHKM rules	446	799
Other (e.g. parse failures)	8	7
Total missing useful phrase pairs	1,994	2,199

Table 4: reasons that ATS-useful phrase pairs could not be extracted by GHKM as phrasal rules

The second reason is that the GHKM model is sometimes forced by its syntactic constraints to include extra words. Sometimes this is only target language words, and this is often useful — the rules are learning to insert these words in their proper context. But most of the time, source language words are also forced to be part of the rule, and this is harmful — it makes the rules less general. This latter case is often due to poorly aligned target language words (such as the 我 in our Section 3 rule extraction example), or unaligned words under large, flat constituents.

Another factor here: some of the phrase pairs are learned by both systems, but GHKM is more specific about the context of use. This can be both a strength and a weakness. It is a strength when the syntactic context helps the phrase to be used in a syntactically correct way, as in

VP(VBD(said) x_0:SBAR-C) $\leftrightarrow$ 说 x_0

where the syntax rule requires a constituent of type SBAR-C. Conversely its weakness is seen when the context is too constrained. For example, ATS can easily learn the phrase

总理 $\leftrightarrow$ prime minister

and is then free to use it in many contexts. But GHKM learns 45 different rules, each that translate this phrase pair in a unique context. Figure 6 shows a sampling. Notice that though many variations are present, the decoder is unable to use any of these rules to produce certain noun phrases, such as "current Japanese Prime Minister Shinzo Abe", because no rule has the proper number of English modifiers.

NPB(NNP(prime) NNP(minister) x_0:NNP) $\leftrightarrow$ x_0 总理
NPB(x_0:NNP NNP(prime) NNP(minister) x_1:NNP) $\leftrightarrow$ x_0 总理 x_1
NPB(x_0:JJ NNP(prime) NNP(minister) x_1:NNP) $\leftrightarrow$ x_0 总理 x_1
NPB(NNP(prime) NNP(minister) x_0:NNP) $\leftrightarrow$ 总理 x_0
NPB(NNP(prime) NNP(minister)) $\leftrightarrow$ 总理
NPB(NNP(prime) NNP(minister) x_0:NNP x_1:NNP) $\leftrightarrow$ x_0 x_1 总理
NPB(x_0:DT x_1:JJ JJ(prime) NN(minister)) $\leftrightarrow$ x_0 x_1 总理
NPB(x_0:NNP NNP(prime) NNP(minister) x_1:NNP) $\leftrightarrow$ x_0 总理 x_1
NPB(x_0:NNP NNP(prime) NNP(minister) x_1:NNP) $\leftrightarrow$ x_0 总理 x_1

Figure 6: a sampling of the 45 rules that translate 总理 to prime minister

5 Coverage Improvements

Each of the models presented so far has advantages and disadvantages. In this section, we consider ideas that make up for deficiencies in the GHKM model, drawing our inspiration from the strong points of the ATS model. We then measure the effects of each idea empirically, showing both what is gained and the potential limits of each modification.

5.1 Composed Rules

Galley et al. (2006) proposed the idea of composed rules. This removes the minimality constraint required earlier: any two or more rules in a parent-child relationship in the derivation tree can be combined to form a larger, composed rule. This change is similar in spirit to the move from word-based to phrase-based MT models, or parsing with a DOP model (Bod et al., 2003) rather than a plain PCFG.

Because this results in exponential variations, a size limit is employed: for any two or more rules to be allowed to combine, the size of the resulting rule must be at most n. The size of a rule is defined as the number of non-part-of-speech, non-leaf constituent labels in a rule's target tree. For example, rules 1-5 shown in Section 3 have a size of 0, and rule 6 has a size of 10. Composed rules are extracted in addition to minimal rules, which means that a larger n limit always results in a superset of the rules extracted when a smaller n value is used. When n is set to 0, then only minimal rules are extracted. Table 5 shows the growth in the number of rules extracted for several size limits.

Size limit (n)	Chinese	Arabic
0 (minimal)	2,487,110	662,037
2	12,351,297	2,742,513
3	26,917,088	4,824,928
4	55,781,061	8,487,656

Table 5: increasing the size limit of composed rules significantly increases the number of rules extracted

In our previous analysis, the main reason that GHKM did not learn translations for ATS-useful phrase pairs was due to its minimal-only approach. Table 6 shows the effect that composed rule extraction has on the total number of ATS-useful phrases missing. Note that as the allowed size of composed rule increases, we are able to extract an greater percentage of the missing ATS-useful phrase pairs.

Size limit (n)	Chinese	Arabic
0 (minimal)	1,994	2,199
2	1,478	1,528
3	1,096	1,210
4	900	1,041

Table 6: number of ATS-useful phrases still missing when using GHKM composed rule extraction

Unfortunately, a comparison of Tables 5 and 6 indicates that the number of ATS-useful phrase pairs gained is growing at a much slower rate than the total number of rules. From a practical standpoint, more rules means more processing work and longer decoding times, so there are diminishing returns from continuing to explore larger size limits.

5.2 SPMT Model 1 Rules

An alternative for extracting larger rules called SPMT model 1 is presented by Marcu et al. (2006). Though originally presented as a separate model, the method of rule extraction itself builds upon the minimal GHKM method just as composed rules do. For each training example, the method considers all source language phrases up to length L. For each of these phrases, it extracts the smallest possible syntax rule that does not violate the alignments. Table 7 shows that this method is able to extract rules that cover useful phrases, and can be combined with size 4 composed rules to an even better effect. Since there is some overlap in these methods, when combining the two methods we eliminate any redundant rules.

Method	Chinese	Arabic
composed alone (size 4)	900	1,041
SPMT model 1 alone	676	854
composed + SPMT model 1	663	835

Table 7: ATS-useful phrases still missing after different non-minimal methods are applied

Note that having more phrasal rules is not the only advantage of composed rules. Here, combining both composed and SPMT model 1 rules, our gain in useful phrases is not very large, but we do gain additional, larger syntax rules. As discussed in (Galley et al., 2006), composed rules also allow the learning of more context, such as

```
ADJP(ADVP(RB(far)
          CC(and)
          RB(away)    ↔ 远远 x0
     x0:JJ)
```

This rule is not learned by SPMT model 1 because it is not the smallest rule that can explain the phrase pair, but it is still valuable for its syntactic context.

5.3 Restructuring Trees

Table 8 updates the causes of missing ATS-useful phrase pairs. Most are now caused by syntactic constraints, thus we need to address these in some way.

GHKM translation rules are affected by large, flat constituents in syntax trees, as in the `prime minister` example earlier. One way to soften this constraint is to binarize the trees, so that wide constituents are broken down into multiple levels of tree structure. The approach we take here is head-out binarization (Wang et al., 2007), where any constituent with more than two children is split into partial constituents. The children to the left of the head word

Category of ATS-useful phrase pairs	Chinese	Arabic
Too large	12	9
Extra target words in GHKM rules	218	27
Extra source words in GHKM rules	424	792
Other (e.g. parse failures)	9	7
Total missing useful phrase pairs	663	835

Table 8: reasons that ATS-useful phrase pairs are still not extracted as phrasal rules, with composed and SPMT model 1 rules in place

are binarized one direction, while the children to the right are binarized the other direction. The top node retains its original label (e.g. NPB), while the new partial constituents are labeled with a bar (e.g. $\overline{\mathrm{NPB}}$). Figure 7 shows an example.

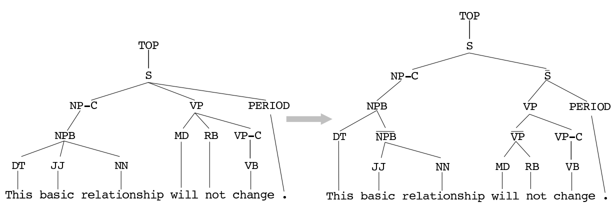

Figure 7: head-out binarization in the target language: S, NPB, and VP are binarized according to the head word

Table 9 shows the effect of binarization on phrasal coverage, using both composed and SPMT rules. By eliminating some of the syntactic constraints we allow more freedom, which allows increased phrasal coverage, but generates more rules.

Category of missing ATS-useful phrase pairs	Chinese	Arabic
Too large	16	12
Extra target words in GHKM rules	123	12
Extra source words in GHKM rules	307	591
Other (e.g. parse failures)	12	7
Total missing useful phrase pairs	458	622

Table 9: reasons that ATS-useful phrase pairs still could not be extracted as phrasal rules after binarization

6 Evaluation of Translations

To evaluate translation quality of each of these models and methods, we ran the ATS decoder using its extracted phrase pairs and the syntax-based decoder using all the rule sets mentioned above. Table 10 describes the development and test datasets used, along with four references for measuring BLEU. Tuning was done using Maximum BLEU hill-climbing (Och, 2003). Features used for the ATS system were the standard set. For the syntax-based translation system, we used a similar set of features.

		# of lines	
	Dataset	Chinese	Arabic
Development set	NIST 2002 MT eval (sentences $<$ 47 tokens)	925	696
Test set	NIST 2003 MT eval	919	663

Table 10: development and test corpora

Table 11 shows the case-insensitive NIST BLEU4 scores for both our development and test decodings. The BLEU scores indicate, first of all, that the syntax-based system is much stronger in translating Chinese than Arabic, in comparison to the phrase-based system. Also, the ideas presented here for improving phrasal coverage generally improve the syntax-based translation quality. In addition, composed rules are shown to be helpful as compared to the minimal runs. This is true even when SPMT model 1 is added, which indicates that the size 4 composed rules bring more than just improved phrasal coverage.

	Chinese		Arabic	
Experiment	Dev	Test	Dev	Test
Baseline ATS	34.94	32.83	50.46	50.52
Baseline GHKM (minimal only)	38.02	37.67	49.34	49.99
GHKM composed size 2	40.24	39.75	50.76	50.94
GHKM composed size 3	40.95	40.44	51.56	51.48
GHKM composed size 4	41.36	40.69	51.60	51.71
GHKM minimal + SPMT model 1	39.78	39.16	50.17	51.27
GHKM composed + SPMT model 1	42.04	41.07	51.73	51.53
With binarization	42.17	41.26	52.50	51.79

Table 11: evaluation results (reported in case-insensitive NIST BLEU4)

7 Conclusions

Both the ATS model for phrase-based machine translation and the GHKM model for syntax-based machine translation are state-of-the-art methods. Each extraction method has strengths and weaknesses as compared to the other, and there are surprising differences in phrasal coverage — neither is merely a superset of the other. We have shown that it is possible to gain insights from the strengths of the phrase-based extraction model to increase both the phrasal coverage and translation accuracy of the syntax-based model.

However, there is still room for improvement in both models. For syntax models, there are still holes in phrasal coverage, and other areas are needing progress, such as decoding efficiency. For phrase-based models, incorporating syntactic knowledge and constraints may lead to improvements as well.

8 Acknowledgments

The authors wish to acknowledge our colleagues at ISI, especially David Chiang, for constructive criticism on an early draft of this document, and several reviewers for their detailed comments which helped us make the paper stronger. We are also grateful to Jens-Sönke Vöckler for his assistance in setting up an experimental pipeline, without which this work would have been much more tedious and difficult. This research was supported under DARPA Contract No. HR0011-06-C-0022.

References

Rens Bod, Remko Scha, and Khalil Sima'an, editors. 2003. *Data-Oriented Parsing*. CSLI Publications, University of Chicago Press.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL 2005*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4).

Brooke Cowan, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proc. EMNLP 2006*.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. ACL 2003*.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proc. ACL 2006*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. HLT-NAACL 2004*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. ACL 2006*.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proc. ACL 2003, companion volume*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proc. AMTA 2006*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT-NAACL 2003*.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. EMNLP 2006*.

I. Dan Melamed. 2004. Statistical machine translation by parsing. In *Proc. ACL 2004*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL 2003*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. ACL 2005*.

Wei Wang, Kevin Knight, and Daniel Marcu. 2007. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proc. EMNLP and CoNLL 2007*.

Dekai Wu and Hongsing Wong. 1998. Machine translation with a stochastic grammatical channel. In *Proc. ACL 1998*.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. ACL 2001*.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proc. NAACL HLT 2006*.