Enriching SCFG Rules Directly From Efficient Bilingual Chart Parsing

Martin Čmejrek, Bowen Zhou, Bing Xiang

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{martin.cmejrek,bzhou,bxiang}@us.ibm.com

Abstract

In this paper, we propose a new method for training translation rules for a Synchronous Context-free Grammar. A bilingual chart parser is used to generate the parse forest, and EM algorithm to estimate expected counts for each rule of the ruleset. Additional rules are constructed as combinations of reliable rules occurring in the parse forest. The new method of proposing additional translation rules is independent of word alignments. We present the theoretical background for this method, and initial experimental results on German-English translations of Europarl data.

1. Introduction

Statistical machine translation has dramatically improved over the last few decades. Phrase-based and syntax-based systems are probably the most commonly adopted approaches. Although they implement various modeling techniques to improve performance on different languages, domains, or user scenarios, they usually share the same basic pattern for generating rules: starting with word alignments, and using heuristic approaches to extract phrase pairs. These approaches are very successful in handling local linguistic phenomena, but handling longer distance dependencies can be more difficult. One of the reasons might be that, to avoid combinatorial explosion, and to achieve reasonable model size, these heuristics usually apply constraints, such as limitations of the phrase length or non-terminal span, sometimes too restrictive to extract some good rules. Another reason is the deterministic nature of those heuristics that does not allow to recover from errors in the word alignment.

In this work, we learn rules for hierarchical phrase based MT systems directly from the parallel data. The main contribution of this paper is a new method for proposing translation rules which is independent of bilingual word alignments.

Let us have an example of a German-English sentence pair from the Europarl corpus [1].

(1) GER: die herausforderung besteht darin diese systeme zu den besten der welt zu machen
ENG: the challenge is to make the system the very best

We can see that the pairs of long sequences (*diese systeme ... der welt, the system ... best*) and (*zu machen, to make*) are swapped. It would be nice to generate rules that can handle long distance reorderings, still with a reasonably low number of terminals, for example:

(2) $X \rightarrow \langle$besteht darin X_1 zu X_2, is to $X_2 X_1\rangle$,

There are 127 sentence pairs out of 300K of the training data that contain this pattern, but this rule was not extracted into the baseline ruleset using the conventional approach [2]: either because of word alignment errors, or because the maximum span for rule extraction is lower than 11 words.

We want to learn new rules by combining existing rule usages. Thus we might combine:

(3) $X \rightarrow \langle$besteht darin, is $\rangle$
$X \rightarrow \langle X_1$ zu X_2, to $X_2 X_1\rangle$

to get the rule (2).

Our approach, as shown in Figure 1, consists of bilingual chart parsing (BCP) of the training data, combining rules found in the chart using a *rule arithmetic* to propose new rules, and using EM to estimate rule probabilities.

The paper is structured as follows: In Section 1, we explain our main motivation, summarize previous work, and briefly introduce the formalism of hierarchical phrase-based translation. In Section 2, we present details of bilingual chart parsing. The mathematical background for EM algorithm is presented in Section 3. In Section 4, we describe additions to the baseline ruleset that extend the grammar coverage during the EM training. The main topic of this work, the method for generating new translation rules is described in Section 5. In Section 6, we present results on German-English translation of Europarl corpus. Finally, we conclude in Section 7.

1.1. Related work

In many previous works, the EM algorithm was used to estimate probabilities of translation rules:

Wu [3] uses EM to directly estimate joint word alignment probabilities of Inversion Transduction Grammar (ITG).

Marcu and Wong [4] use EM to estimate joint phrasal translation model (JPTM). The translation process is described as simultaneous generation of both languages. First, a bag of bilingual phrase pairs, so-called *concepts* is generated, then these phrases are permuted to create a sentence pair. The search space of this method is huge, so Birch et

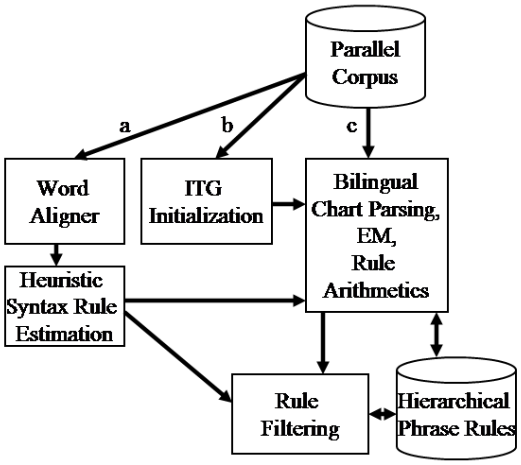

Parallel Corpus
a
b
c
Word Aligner
ITG Initialization
Bilingual Chart Parsing, EM, Rule Arithmetics
Heuristic Syntax Rule Estimation
Rule Filtering
Hierarchical Phrase Rules

Figure 1: *Enriching SCFG rules from bilingual chart parsing.*

al. [5] reduce it by using only concepts that match the high-confidence GIZA++ alignments. Similarly, Cherry and Lin [6] use ITG for pruning.

May and Knight [7] use EM algorithm to train tree-to-string rule probabilities, and use the Viterbi derivations to re-align the training data.

Huang and Zhou [8] use EM to estimate conditional rule probabilities $P(\alpha|\gamma)$ and $P(\gamma|\alpha)$ for Synchronous Context-free Grammar. We further improve their approach by using additional rules in the bilingual parsing and EM training.

Galley et al. [9] define minimal rules for tree-to-string translation, and (similarly to our *rule arithmetic*) merge them into composed rules. The EM is used to estimate rule weights. While in their method, word alignments are used to define all rules, our method proposes new rules independently of word alignments.

1.2. Formally syntax-based models

Our baseline model follows the Chiang's hierarchical model [2, 10, 11] based on Synchronous Context-free Grammar. The rules have form

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle, \quad (4)$$

where X is the only non-terminal in the grammar, γ and α are source and target strings with both terminals and non-terminals, subject to the constraint that there is always a one-to-one correspondence $\sim$ between those non-terminals. The $\sim$ is often represented by co-indexing corresponding non-terminals. Rules with terminals only are called *phrasal* rules, while rules with non-terminals are *abstract* rules. We limit the number of non-terminals in each rule to no more than two, thus ensuring the rank of SCFG is two. The set of rules, denoted as $\mathcal{R}$, are automatically extracted from a parallel corpus [2, 10] with word-alignments obtained from GIZA++ [12]. Finally, an implicit *glue* rule is embedded with decoder to allow for translations that can be achieved by sequentially linking sub-translations generated chunk-by-chunk:

$$X \rightarrow \langle X_1 X_2, X_1 X_2 \rangle. \quad (5)$$

X is also the start symbol.

All rules in $\mathcal{R}$ are paired with statistical parameters (i.e., weighted SCFG), which combines with other features to form the models using a log-linear framework. The decoder tries to maximize:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times \\ \prod_i \prod_{X \rightarrow <\gamma, \alpha> \in D} \phi_i(X \rightarrow <\gamma, \alpha>)^{\lambda_i}, \quad (6)$$

where the set of $\phi_i(X \rightarrow <\gamma, \alpha>)$ are features defined over given production rule, and $P_{LM}(e)$ is the language model score on hypothesized output, the λ_i is the feature weight.

The baseline model follows Chiang's hierarchical model [2]: conditional probabilities $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$; lexical weights [13] $P_w(\gamma|\alpha)$ and $P_w(\alpha|\gamma)$; word counts $|e|$; rule counts $|D|$; abstraction penalty (to account for the accumulated number of non-terminals in D); target n-gram language model $P_{LM}(e)$; and the glue rule penalty to learn preference of non-terminal rewriting over serial combination through Eq. (5).

We note, however, that these parameters are often poorly estimated due to the scarceness of data and the usage of inaccurate heuristics. We try to alleviate this problem by EM training in Section 3.

2. Bilingual chart parsing

In this section, we describe the algorithm for bilingual parsing and present our implementation details necessary for a substantial speedup of the original algorithm [8], allowing for model extensions further mentioned in Section 4.

In bilingual parsing, we are interested in finding all synchronous derivations Φ of the sentence pair (e_1^M, f_1^N) using rules from $\mathbf{R}$. A bilingual chart parser, a parallel version of a CYK parser is capable of doing it. The algorithm is described in Fig. 2.

```
$RSpans := precompute(R, \mathbf{e}, \mathbf{f})$
for $i, j, k, l$ in bottom-up order, such that
                $1 \leq i \leq j \leq M,$
                $1 \leq k \leq l \leq N$
    for $\rho \in RSpans(i, j, k, l)$
        switch $\rho.n$
        case 0:
            $t_{ijkl}.push(\rho)$
        case 1:
            if $(filled(t_{\rho.bp1}))$
                $t_{ijkl}.push(\rho)$
        case 2:
            if $(filled(t_{\rho.bp1}) \& filled(t_{\rho.bp2}))$
                $t_{ijkl}.push(\rho)$
```

Figure 2: Bilingual chart parser for SCFG.

The chart T is a set of cells t_{ijkl} such that $1 \leq i \leq j \leq M$ and $1 \leq k \leq l \leq N\}$. Each cell t_{ijkl} represents all parses with span (i, j, k, l), i.e. over the pair of subsequences (e_i^j, f_k^l). Line 2 iterates all cells t_{ijkl} in a bottom-up order, thus it is granted that parses for all subspans have been computed before visiting t_{ijkl}. In each step, we try to apply all rules $r \in R$ that could generate (e_i^j, f_k^l). In our implementation, we use the structure $RSpans_{ijkl}$ to provide access to all rules with matching terminal sequences, and to appropriate non-terminal spans. In the figure, we use ρ as a “syntactic sugar”: $\rho.r$ denotes the synchronous rule, $\rho.n$ is the number of non-terminals in the rule, and $\rho.bp1 = (i', j', k', l')$ and $\rho.bp2 = (i'', j'', k'', l'')$ are spans of the non-terminals.

If eventual non-teminal spans are filled, i.e. $t_{\rho.bp1}$ and $t_{\rho.bp2}$ are not empty, it means that $\rho.r$ can generate (e_i^j, f_k^l) and we add this hypothesis to the chart cell t_{ijkl}.

Finally, if the root cell $t_{1,M,1,N}$ is not empty, we know that the sentence pair can be generated by the ruleset R, and all derivations can be accessed from the chart by following the backpointers $\rho.bp_i$.

The implementation of the loop on line 5 of algorithm 2 is critial for the system speed. It would be very inefficient to try all rules from R and all possible non-terminal spans. Fortunately, it is possible to construct source and target prefix trees to represent all sequences of terminals and up to 2 non-terminals (while remembering their spans) for any given sentence pair, remembering the non-terminal spans. Thus it is possible to quickly retrieve all the rules relevant to the sentence pair and to access them efficiently through the data structure $RSpans$.

3. Estimating rule probabilities

Let C be a training corpus of sentence pairs $\mathbf{e} = e_1^M$ and $\mathbf{f} = f_1^N$ of source and target sentences.

For each sentence pair $\mathbf{e}, \mathbf{f}$, the ’E’ step of the EM algorithm will enumerate all possible derivations Φ, and will calculate the expected count $c(r)$ that each rule r was used to produce the corpus C:

$$c(r) = \sum_{\mathbf{e},\mathbf{f} \in C} \sum_{\phi \in \Phi} P(r, \phi | \mathbf{e}, \mathbf{f}), \tag{7}$$

where $P(r, \phi | \mathbf{e}, \mathbf{f})$ is the probability of the rule r in the derivation ϕ given the sentence pair $\mathbf{e}, \mathbf{f}$. The expected counts are then used in the ’M’ step to update and normalize rule probabilities:

$$P(r) = \frac{c(r)}{\sum_{r' \in R: L(r') = L(r)} c(r')}, \tag{8}$$

where $L(r)$ denotes the left-hand side of the rule r. The implementation is trivial in our case, since the set of left-hand sides is $\{X\}$.

The expected counts can be computed using inside probabilities $\beta_{ijkl}(X)$ and outside probabilities $\alpha_{ijkl}(X)$ defined as follows:

$$\beta_{ijkl}(X) = P(X \Rightarrow^* e_i^j; f_k^l) \tag{9}$$

$$\alpha_{ijkl}(X) = P(S \Rightarrow^* e_1^{i-1}, X, e_{j+1}^M; f_1^{k-1}, X, f_{l+1}^N) \tag{10}$$

In other words, the $\beta_{ijkl}(X)$ represents the probability of deriving the two parallel sequences e_i^j and f_k^l from X, while the $\alpha_{ijkl}(X)$ is the probability of all derivations of the remaining parts of the sentence pair $\mathbf{e}, \mathbf{f}$, which are not spanned by X. Since there is only one non-terminal symbol X, we can omit it from the following text.

The inside probabilities from the Eq. (9) can be also defined recursively as:

$$\beta_{ijkl} = \sum_{\rho \in t_{ijkl}} P(\rho.r) \prod_{(i'j'k'l') \in \rho.bp} \beta_{i'j'k'l'}, \tag{11}$$

and they can be computed dynamically during the chart parsing in Fig. 2. Since both algorithms visit the chart cells in the same ordering, it is actually preferable to parse and compute the inside probabilities in one turn. The inside probability can also be used as a measure for pruning low probability hypotheses.

The outside probabilities can be computed recursively by iterating the chart in top-down ordering, starting from the root cell

$$\alpha_{1,M,1,N} := 1, \tag{12}$$

and propagating the probability mass as

$$\alpha_{\rho.bp1} += P(\rho.r)\alpha_{ijkl} \tag{13}$$

for hypotheses with one non-terminal, and

$$\alpha_{\rho.bp1} += P(\rho.r)\alpha_{ijkl}\beta_{\rho.bp2} \tag{14}$$

$$\alpha_{\rho.bp2} += P(\rho.r)\alpha_{ijkl}\beta_{\rho.bp1} \tag{15}$$

for hypotheses with two non-terminals.

Finally, the contributions to the rule expected counts are computed as

$$c(\rho.r) += \frac{P(\rho.r)\alpha_{ijkl} \prod_{i=1}^{\rho.n} \beta_{\rho.bp_i}}{\beta_{1,M,1,N}}, \tag{16}$$

and probabilities $P(\rho)$ normalized using Eq. (8).

In general, EM algorithm prefers shorter derivations with longer rules, since the derivation probability is the product of rule probabilities. We take the modeling approach similar to [7] and normalize $P(r)$ by $c_{size}(s)^{s-1}$, where s is the total number of terminals on source and target sides of the r, and c_{size} corresponds to the distribution of rule lengths in the training data.

4. Improving the grammar coverage

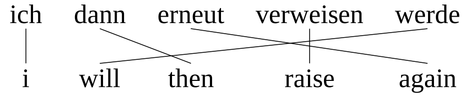
When trying to parse the EUROPARL corpus, we realized that many sentence pairs cannot be parsed. Depending on the method of phrase extraction, the number of unparsable sentences varied from 70% for the *union* type of bilingual alignment to 20% for *grow-diag-final*. The reasons are: structural complexity, OOV translation pairs, and liberal translations in the training data.

Let us have an example of a German-English sentence pair:

(17) GER: meine frage betrifft eine angelegenheit die am donnerstag zur sprache kommen wird und auf die ich dann erneut verweisen werde
ENG: my question relates to something that will come up on thursday and which i will then raise again

with the following alignment between the ends of the sentences:

(18)



We can see two examples of swaps–between *erneut* and *verweisen*, and between *dann erneut verweisen* and *werde*. It can be seen that either the rule $X \rightarrow \langle$erneut X_1, X_1 again$\rangle$ or $X \rightarrow \langle X_1$verweisen,raise $X_1\rangle$ is needed to parse this sentence pair using short rules. If neither of the two rules were extracted to the ruleset (because of pruning or word alignment error), the sentence pair cannot be parsed.

Another problem we noticed is that some sentence pairs have a very scarce parse forest, consisting of many "bad" rules, while the "good" rules are ignored.

(19) GER: **nach monatelangen und** weltweiten konsultationen wird nun im donaldson bericht die ausweitung dieser forschungen zu therapeutischen zwecken empfohlen
EN: **and after consulting world wide for many months** the donaldson report recommends extending such research for therapeutic purposes

The Example (19) shows a common pattern of errors that can be observed in the data: An error in word alignment of low frequency words (here, the less frequent variant *world wide* was not aligned with *weltweiten*) results in extraction of an asymmetric phrase pair ($X \rightarrow \langle$nach monatelangen und,and after ... many months$\rangle$). This phrase pair can often be the only rule spanning the low frequency word. In addition, during the EM training on the same sentence pair, the parser is forced to use another asymmetric rule to counterbalance (by covering *weltweiten*) this error instead of using good rules, such as $X \rightarrow \langle$konsultationen,consulting$\rangle$. As a result, the EM accumulates expected counts for asymmetric rules instead of good rules. We could better recover from this alignment error if we could delete either *weltweiten* and *world wide*, or *weltweiten konsultationen*.

Inspired by ITG [3], we extended our ruleset by the following rules, that will provide "backoff" parses and scoring for the SCFG rules:

(20) $\langle X_1, X_1 f\rangle$, $\langle X_1, f X_1\rangle$, $\langle X_1 e, X_1\rangle$, $\langle e X_1, X_1\rangle$,

(21) $\langle X_1 X_2, X_2 X_1\rangle$.

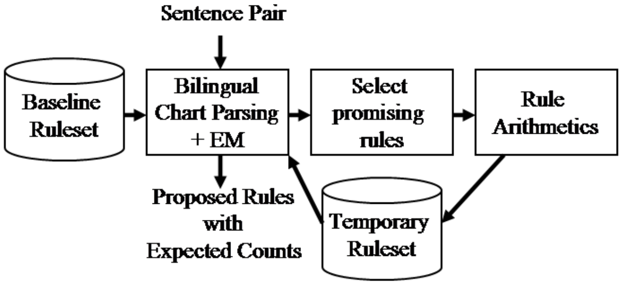


Figure 3: *Proposing new rules from BCP.*

Rules (20) enable insertions and deletions, while rule (21) allows for aligning swapped constituents.

Since insertions and deletions are represented as abstract rules, they can be applied only at fringes of already parsed spans. In addition, we require that the X_1 non-terminal of a deletion rule does not span more than 4 words, thus we prune parses that would delete large sequences of words.

In this paper, we use the term *ITG rules* for these rules, although we are aware of the difference from the original ITG definition.

After adding the new types of rules, only 0.08% of sentences still cannot be parsed. We found that those sentences are usually misaligned or they contain incomplete translations, and skipping them is the best option.

5. Proposing new rules with rule arithmetic

The main idea of this work is to propose new rules independently of the bilingual word alignments as shown in the Fig. 3. We parse each sentence pair using the baseline ruleset extended by the new rule types (20) and (21). Then we select the *most promising* rule usages and combine each two of them using the *rule arithmetic* to propose new rules. We put the new rules into a temporary pool, and parse and compute probabilities and expected counts again, this time we use rules from the baseline and from the temporary pool. Finally, we dump expected counts for proposed rules, and empty the temporary pool. This way we can try to propose many rules for each sentence pair, and to filter them later using accumulated expected counts from the EM.

The term *most promising* is purposefully vague—to cover all possible approaches to filtering rule usages. In our implementation, we are limited by space and time, and we have to prune the number of rules that we can combine. We use expected counts as the main scoring criterion. When computing the contributions to expected counts from particular rule usages as described by (16), we remember the n-best contributors, and use them as candidates after the expected counts for the given sentence pair are estimated.

The *rule arithmetic* used to combine rules defines the operation of *addition*; the main idea is shown in examples in Fig. 4:

First, create span projections for both source and target sides of both rules. Use symbol 0 for all unspanned positions,

	Training	Tune	Test
Sentence pairs	296,999	1,000	1,000
German tokens	4,210,289	14,179	14,055
English tokens	4,601,538	15,530	15,280

Table 1: Training and test data for German-English translation

copy terminal symbols as they are, and use symbols -1, -2, -3, and -4 to transcribe X_1 and X_2 from the first rule, and X_1 and X_2 from the second rule. Repeat the non-terminal symbol on all spanned positions. In each example in Fig. 4, lines 1 show the positions in the sentence, lines 2 and 3 show the rule span projections of the two rules.

Second, merge source span projections (lines 4), record mappings of non-terminal symbols. We require that merged projections are *continuous*. In Example (22), two short phrasal rules are being merged to produce a longer phrasal rule. On the other hand, merging discontinuous rules as in Example (23) is not defined.

New abstract rules can be created (among others) by merging terminal sequences with either a glue or swap rule, as in (24). We allow substituting non-terminal symbols by terminals, but we require that the whole span of the non-terminal is fully replaced. In other words, shortenings of non-terminal spans are not allowed.

Since insertion and deletion rules are represented as abstract rules with one non-terminal, we can use the same pattern as in example (24) to combine them with other rules. We restrict combinations with other insertions, deletions, glues or swaps.

The Example (25) is our motivation example generated as a combination of a phrasal rule and an abstract rule.

Example (26) is a combination of two abstract rules, which maps the non-terminals X_1 of the second rule into X_2.

Finally, the Example (27) is the most difficult. The non-terminal span -1 (representing X_1 of the first rule) is replaced by source and target terminals *diese* and *the* and by a non-terminal sequence of -3. Note that none of the non-terminal spans is shortened. The remaining span sequences -3 and -2 are then mapped to X_1 and X_2.

Third, collect new rule. The merged rule usages (lines 5) are then generalized into rules, so that they are not limited to the particular span for which they were originally proposed.

6. Experiments

In the following, we describe the data used for our experiments, the training framework, and we present results.

6.1. Data

We carried out our experiments with German-to-English translation using the data from the Europarl [1] corpus. The amounts of training, tuning and testing data are listed in Table 1.

The data is a filtered subset, with focus on travel domain. Our motivation is to improve speech-to-speech translation, thus all punctuation was removed from the data, and the text was converted to lower case.

6.2. Training framework

Our training framework is shown in Fig. 1. The baseline ruleset is obtained from word-level alignments by the baseline heuristic approach (a).

Additional ITG rules (b) are generated from the training data: deletion rules for all source words, insertion rules for all target words, and 1-1 terminal rules for all co-occurring word pairs.

The baseline and ITG rules are combined and filtered, so that baseline rules are preferred, while rules only occurring in the ITG part have much higher costs and serve only as a backoff to increase the parsability of the training data. Also, the ITG-only rules are **not** used for decoding.

The bilingual chart parsing (c) estimates rule probabilities by EM, and uses rule arithmetic to propose rules that better explain the training data. The modular architecture allows for different training setups. In the first setup (*EM-costs*), only rule probabilities/costs of abstract rules were estimated by 10 iterations of EM. In the second setup (*EM-propose*), the new rules were proposed after each iteration of EM. In the third setup (*EM-propose&costs*), the proposed rules were merged with the baseline and ITG rules and the rule costs were estimated by EM.

In the following, we are presenting more details about the three setups and present the BLEU scores [14] in the Table 2.

6.3. Baseline

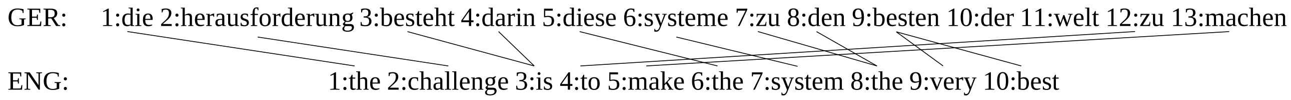
The baseline in our experiments is a formal syntax-based translation model [11]. We train GIZA++ [15] word alignments on the sentence-aligned data, and extract phrase pairs using heuristics grow-diag-final [16]. The phrases were up to 6 and 8 words long on the source and target sides, respectively. The method of extracting abstract rules is similar to [2]. The log-linear model combines 9 features, as described in Section 1.2.

6.4. Using BCP and EM to estimate rule costs

In the first experiment, we were trying to better estimate features of the baseline abstract rules.

As discussed in Section 4, to increase the parsability of the corpus we had to provide additional ITG rules. We added all word pairs that had an entry in at least one of the GIZA++ tables of lexical translation probabilities, but were not present in the baseline phrase table. Then we added deletion and insertion rules for each word from the source and target vocabularies, and finally, we added the glue and swap rules. The total number of ITG rules was less than 1M.

We started from the baseline ruleset extended by the ITG rules, and used EM algorithm to estimate joint probabilities


GER: 1:die 2:herausforderung 3:besteht 4:darin 5:diese 6:systeme 7:zu 8:den 9:besten 10:der 11:welt 12:zu 13:machen
ENG: 1:the 2:challenge 3:is 4:to 5:make 6:the 7:system 8:the 9:very 10:best

(22) Addition $\langle 5,5,0,0,0,0\rangle$ $\langle 6,6,0,0,0,0\rangle$ $X \rightarrow \langle$diese, the$\rangle$
$\langle 6,6,0,0,0,0\rangle$ $\langle 7,7,0,0,0,0\rangle$ $X \rightarrow \langle$systeme, system$\rangle$

1:	...	4	5	6	7	...	...	5	6	7	8	...
2:	...	0	diese	0	0	...	...	0	the	0	0	...
3:	...	0	0	systeme	0	...	...	0	0	system	0	...
4:	...	0	diese	systeme	0	...	...	0	the	system	0	...

5: $\langle 5,6,0,0,0,0\rangle$ $\langle 6,7,0,0,0,0\rangle$ $X \rightarrow \langle$diese systeme, the system$\rangle$

(23) * Addition $\langle 1,1,0,0,0,0\rangle$ $\langle 1,1,0,0,0,0\rangle$ $X \rightarrow \langle$die, the$\rangle$
$\langle 6,6,0,0,0,0\rangle$ $\langle 7,7,0,0,0,0\rangle$ $X \rightarrow \langle$systeme, system$\rangle$

1:	1	2	...	5	6	7	1	2	...	5	6	7	8	...
2:	die	0	...	0	0	0	the	0	...	0	0	0	0	...
3:	...	0	...	0	systeme	0	0	0	...	0	0	system	0	...
4:	die	0	...	0	systeme	0	the	0	...	0	0	system	0	...

5: *undefined*

(24) Addition $\langle 12,13,0,0,0,0\rangle$ $\langle 4,5,0,0,0,0\rangle$ $X \rightarrow \langle$zu machen, to make$\rangle$
$\langle 5,13,5,11,12,13\rangle$ $\langle 4,10,6,10,4,5\rangle$ $X \rightarrow \langle X_1\ X_2,\ X_2\ X_1\rangle$

1:	...	5	...	11	12	13	...	3	4	5	6	...	10
2:	...	0	...	0	zu	machen	...	0	to	make	0	...	0
3:	...	-3	...	-3	-4	-4	...	0	-4	-4	-3	...	-3
4:	...	-3	...	-3	zu	machen	...	0	to	make	-3	...	-3

5: $\langle 5,13,5,11,0,0\rangle$ $\langle 4,10,6,10,0,0\rangle$ $X \rightarrow \langle X_1$ zu machen, to make $X_1\rangle$

(25) Addition $\langle 3,4,0,0,0,0\rangle$ $\langle 3,3,0,0,0,0\rangle$ $X \rightarrow \langle$besteht darin, is$\rangle$
$\langle 5,13,5,11,13,13\rangle$ $\langle 4,10,5,5,6,10\rangle$ $X \rightarrow \langle X_1$ zu X_2, to $X_2 X_1\rangle$

1:	1	2	3	4	4	5	...	11	12	13	1	2	3	4	5	6	...	10
2:	0	0	besteht	darin	0	0	...	0	0	0	0	0	is	0	0	0	...	0
3:	0	0	0	0	-3	-3	...	-3	zu	-4	0	0	0	to	-3	-4	...	-4
4:	0	0	besteht	darin	-3	-3	...	-3	zu	-4	0	0	is	to	-3	-4	...	-4

5: $\langle 3,13,5,11,13,13\rangle$ $\langle 3,10,5,5,6,10\rangle$ $X \rightarrow \langle$besteht darin X_1 zu X_2, is to $X_2 X_1\rangle$

(26) Addition $\langle 3,13,5,13,0,0\rangle$ $\langle 3,10,4,10,0,0\rangle$ $X \rightarrow \langle$besteht darin X_1, is $X_1\rangle$
$\langle 1,2,1,1,0,0\rangle$ $\langle 1,2,1,1,0,0\rangle$ $X \rightarrow \langle X_1$ herausforderung, X_1 challenge$\rangle$

1:	1	2	3	4	5	...	13	1	2	3	4	...	10
2:	0	0	besteht	darin	-1	...	-1	0	0	is	-1	...	-1
3:	-3	herausforderung	0	0	0	...	0	-3	challenge	0	0	...	0
4:	-3	herausforderung	besteht	darin	-1	...	-1	-3	challenge	is	-1	...	-1

5: $\langle 1,13,1,1,5,13\rangle$ $\langle 1,10,1,1,4,10\rangle$ $X \rightarrow \langle X_1$ herausforderung besteht darin X_2, X_1 challenge is $X_2\rangle$

(27) Addition $\langle 5,13,5,11,13,13\rangle$ $\langle 4,10,6,10,5,5\rangle$ $X \rightarrow \langle X_1$ zu X_2, to $X_2\ X_1\rangle$
$\langle 5,11,6,11,0,0\rangle$ $\langle 6,10,7,10,0,0\rangle$ $X \rightarrow \langle$diese X_1, the $X_1\rangle$

1:	...	4	5	6	...	11	12	13	3	4	5	6	7	...	10
2:	...	0	-1	-1	...	-1	zu	-2	0	to	-2	-1	-1	...	-1
3:	...	0	diese	-3	...	-3	0	0	0	0	0	the	-3	...	-3
4:	...	0	diese	-3	...	-3	zu	-2	0	to	-2	the	-3	...	-3

5: $\langle 5,13,6,11,13,13\rangle$ $\langle 4,10,7,10,5,5\rangle$ $X \rightarrow \langle$diese X_1 zu X_2, to X_2 the $X_1\rangle$

Figure 4: Rule arithmetic – addition

$P(\gamma, \alpha)$ of all rules. The initial joint probabilities were set based on co-occurrences. The rule probabilities $P(r)$ needed for scoring during the 'E' step are combinations of the joint probability $P(\gamma, \alpha)$, conditional probabilities $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$, and lexical weights $P_w(\gamma|\alpha)$ and $P_w(\alpha|\gamma)$.

In the 'M' step, we re-estimated the joint probability $P(\gamma, \alpha)$. The joint probability itself was not used as a feature in the decoder. Instead, we used it to update the conditional probabilities $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$ for abstract rules. The phrasal rules used the baseline values of $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$.

Since insertions would overgenerate the decoder output and deletions would degrade the translation performance, the ITG rules were finally removed from the model before tuning and decoding.

The results of this experiment are marked as *EM-costs*. After the first iteration, we were able to improve by 0.68 BLEU on the testset. The improvement in the next iterations varies from 0.38 to 0.86 BLEU points.

6.5. Using BCP and EM to propose new rules

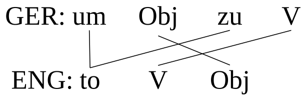
In the next experiment, we used the model resulting from the first iteration of EM models with EM-trained probabilities from the previous experiment. Then we used the approach described in Section 5 to propose new rules.

After parsing the entire training set, we collected 9M (non-unique) proposed new rules. This number is too big, and also the quality varies, some precise filtering is necessary. As the first, we wanted to avoid the problem of overfitting that could be caused by selecting long rules, thus we ignored all proposed rules with more than 8 terminals on either side. In order to get rid of noisy rules, often coming from low quality sentence pairs, we selected only those rules that were proposed from at least 2 sentence pairs. Finally, we sorted the rest of the rules by their expected counts, and selected 100K best rules. We used the expected counts to estimate the rules joint and conditional probabilities.

The Table 3 presents a sample of new rules proposed during this experiment. The table is divided into three parts, presenting rules from the top, middle, and bottom of the 100K list. The quality of the rules is high even in the middle part of the table, the tail part is worse.

We were surprised by seeing short rules consisting of frequent words. For example *um* X_1 - *in order* X. When looking into word-level alignments, we realized that these rules following the pattern prevent the baseline approach from extracting the rule.

(28)



Similarly many other rules match the pattern of beginning of a subordinated clause, such as *that is why*, or insertions, such as *of course*, which both have to be strictly followed by VSO construction in German, in contrast to the SVO word order in English.

	devset	testset
baseline		
	23.852	25.447
EM - costs		
iteration 0	24.394	26.122
1	24.365	25.826
2	24.433	25.936
3	24.375	26.047
4	24.409	25.936
5	24.300	26.259
6	24.339	26.197
7	23.985	25.827
8	24.129	26.305
9	23.988	25.940
10	24.079	26.226
EM - propose		
iteration 0	24.418	26.122
EM - propose & costs		
iteration 0	24.837	26.408

Table 2: BLEU scores on Europarl testset

Finally, we merged the new proposed rules with the baseline rules. The result of this experiment is marked as *EM-propose*. The improvement is 0.68 BLEU points over the baseline.

6.6. Using BCP and EM to estimate rule costs for proposed rules

The third experiment is a combination of the previous two. We merged the new proposed rules with the baseline ruleset and with ITG rules. Then we estimated rule costs the same way as in the first experiment. We hope that this experiment will help to further distinguish the bad rules added in the previous step from the good ones. The experiment is marked *EM-propose&costs*. The gain from the combined approaches is 0.96 BLEU.

7. Conclusion

In this work, we proposed a new approach to the translation rule extraction. We introduced algorithms for efficient bilingual parsing, estimating rule probabilities, and finally, we presented a novel method for synthesizing new rules from the most confident rules within the parse forest. The method does not use bilingual alignments for learning new rules, and is complementary to heuristic alignment-based approaches.

We discussed possible reasons why the new method of learning rules may outperform the baseline method of rule extraction, especially for a language pair with a different word order, such as German and English.

We also showed on experiments that each of the two methods significantly improves the baseline. The improvement is additive, if the two methods are combined.

1 ...	
um X_1	in order X_1
natuerlich X_1	of course X_1
deshalb X_1	this is why X_1
X_1 zu koennen	to X_1
X_1 ist	it is X_1
nach der tagesordnung folgt die X_1	the next item is the X_1
herr X_1 herr kommissar X_2	mr X_1 commissioner X_2
die X_1 der X_2	X_1 the X_2
im gegenteil X_1	on the contrary X_1
nach der tagesordnung folgt X_1	the next item is X_1
X_1 die X_2	the X_1 the X_2
die X_1 die	the X_1
ausserdem X_1	in addition X_1
daher X_1	that is why X_1
wir X_1 nicht X_2	we X_1 not X_2
die X_1 der X_2	the X_2 X_1
deshalb X_1	for this reason X_1
um X_1 zu X_2	to X_2 X_1
X_1 nicht X_2 werden	X_1 not be X_2
... 50001	
nach der tagesordnung folgt die X_1 ueber	the next item is X_1 on
das X_1	X_1 that for
wird X_1 zur X_2	X_1 to X_2
das parlament nimmt den entwurf einer legislativen	parliament adopted the draft legislative
sehr	it is very
X_1 und herren abgeordneten X_2	X_1 and gentlemen X_2
ich habe nicht X_1	i do not have X_1
es X_1	there X_1 the
die X_1 von lissabon	the lisbon X_1
X_1 dabei	in this X_1
frau kommissarin X_1 moechte X_2	commissioner X_1 would like to X_2
... 99991	
X_1 genehmigt	X_1 approved the
X_1 koennen nicht	X_1 cannot have
diese	for them
auch	also needs
dass sich	believe
vorgeschlagen	have proposed
sie X_1 ein	you X_1
es ist	they are
diese vorschlaege gestimmt	voted in favour of these proposals
X_1 rechnungshof	X_1 auditors

Table 3: Sample rules proposed from BCP and EM.

We understand that the results presented here must be verified in other follow-up experiments, and on more language pairs.

8. Acknowledgements

This work is partially supported by the DARPA TRANSTAC program under the contract number NBCH2030001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

9. References

[1] P. Koehn, “Europarl: A parallel corpus for statistical machine translation.” in *Proceedings of MT Summit*, 2005.

[2] D. Chiang, “Hierarchical phrase-based translation,” *Comput. Linguist.*, vol. 33, no. 2, pp. 201–228, 2007.

[3] D. Wu, “Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora,” *Computational Linguistics*, vol. 23, 1995.

[4] D. Marcu and W. Wong, “A phrase-based, joint probability model for statistical machine translation,” in *Proceedings of EMNLP’02*, 2002.

[5] A. Birch, C. Callison-Burch, M. Osborne, and P. Koehn, “Constraining the phrase-based, joint probability statistical translation model,” in *Proceedings on WSMT’06*, 2006, pp. 154–157.

[6] C. Cherry, “Inversion transduction grammar for joint phrasal translation modeling,” in *NAACL-HLT’07/SSST’07*, 2007.

[7] J. May and K. Knight, “Syntactic re-alignment models for machine translation,” in *Proceedings of EMNLP-CoNLL’07*, 2007, pp. 360–368.

[8] S. Huang and B. Zhou, “An EM algorithm for SCFG in formal syntax-based translation,” in *Proc. IEEE ICASSP’09*, 2009, pp. 4813–4816.

[9] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, “Scalable inference and training of context-rich syntactic translation models,” in *Proc. of ACL*, 2006, pp. 961–968.

[10] D. Chiang, “A hierarchical phrase-based model for statistical machine translation,” in *ACL’05*, 2005, pp. 263–270.

[11] B. Zhou, B. Xiang, X. Zhu, and Y. Gao, “Prior derivation models for formally syntax-based translation using linguistically syntactic parsing and tree kernels,” in *Proceedings of the ACL’08: HLT SSST-2*, 2008, pp. 19–27.

[12] F. J. Och and H. Ney, “Improved statistical alignment models,” in *Proc. of ACL*, Hong Kong, China, October 2000, pp. 440–447.

[13] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proc. NAACL/HLT*, 2003.

[14] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: a method for automatic evaluation of machine translation,” Technical Report RC22176, IBM T. J. Watson Research Center, 2001.

[15] F. J. Och, C. Tillmann, and H. Ney, “Improved alignment models for statistical machine translation,” in *Proc. of EMNLP/VLC’99*, MD, USA, 1999, pp. 20–28.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation.” in *ACL*, 2007.