AN EM ALGORITHM FOR SCFG IN FORMAL SYNTAX-BASED TRANSLATION

Songfang Huang[1,2], *Bowen Zhou*[1]

[1]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
[2]Centre for Speech Technology Research, University of Edinburgh, UK
s.f.huang@ed.ac.uk, zhou@us.ibm.com

ABSTRACT

In this paper, we investigate the use of bilingual parsing on parallel corpora to better estimate the rule parameters in a formal syntax-based machine translation system, which are normally estimated from the inaccurate heuristics. We use an Expectation-Maximization (EM) algorithm to re-estimate the parameters of synchronous context-free grammar (SCFG) rules according to the derivation knowledge from parallel corpora based on maximum likelihood principle, rather than using only the heuristic information. The proposed algorithm produces significantly better BLEU scores than a state-of-the-art formal syntax-based machine translation system on the IWSLT 2006 Chinese to English task.

Index Terms— Formal Syntax-based Translation, SCFG, Expectation-Maximization, Inside-Outside Algorithm

1. INTRODUCTION

In recent years, syntax-based translation systems have shown better translation performance than phrase-based systems. One example is the hierarchical phrased-based translation model by Chiang [1] , which uses the synchronous context-free grammar (SCFG) to automatically extract hierarchical structures of natural language. To make it distinct from those syntax-based models that rely on linguistic theory and annotations, we refer to this automatic grammar induction approach as a *formal* syntax-based translation [1, 2].

An SCFG is a synchronous rewriting system generating source and target side string pairs simultaneously based on a context-free grammar. Each synchronous production (i.e., rule) rewrites a non-terminal into a pair of strings, γ and α, with both terminals and non-terminals in both languages. In particular, formal syntax-based models explore hierarchical structures of natural language and utilize only a unified non-terminal symbol X in the grammar:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \tag{1}$$

where $\sim$ is the one-to-one correspondence between X's in γ and α. In this paper we are interested in the estimation of parameters $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$ for SCFG rules.

The set of rules, denoted as $\mathcal{R}$, are automatically extracted from a sentence-aligned parallel corpus [1]. First, bidirectional word-level alignment is carried out on the parallel corpus running GIZA++ [3]. Next, bilingual phrase pairs consistent with word alignments are extracted from the union of bidirectional word-level alignments [4]. Specifically, any pair of consecutive sequences of words below a maximum length is considered to be a phrase pair if its component words are aligned only within the phrase pair and not to any words outside. The resulting bilingual phrase pair inventory is denoted as $\mathcal{BP}$. Each phrase pair PP $\in \mathcal{BP}$ is represented as a production rule $X \rightarrow \langle f_i^j, e_k^l \rangle$, which we refer to as *phrasal rules*. The SCFG rule set encloses all phrase pairs, i.e., $\mathcal{BP} \subset \mathcal{R}$. Next, we loop through each phrase pair PP and generalize the sub-phrase pair contained in PP, denoted as SP_e and SP_f subject to $\text{SP} = (\text{SP}_f, \text{SP}_e) \in \mathcal{BP}$, with co-indexed non-terminal symbols. We thereby obtain a new rule. We will hereafter refer to rules with non-terminal symbols as *abstract rules*.

It is not straightforward to estimate the parameters for abstract rules, because we have not observed the derivations, and therefore we do not know actually how many times each rule has been seen. Chiang [1] uses heuristics to hypothesize a distribution of possible rules as though we observed them in the training data. A count of one is assigned to each initial phrase pair occurrence, which is then equally distributed among the rules derived by subtracting subphrases from it. Treating this distribution as the observed data, relative-frequency estimation is used to obtain rule parameters $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$.

We note that, however, these parameters for abstract rules are often poorly estimated, due to the usage of inaccurate heuristics. In this work, we propose an improved formal syntax-based model, by using the EM algorithm to re-estimate the SCFG rule parameters (EM-SCFG). Rather than using the heuristic information, we instead use knowledge obtained by bilingual parsing of parallel corpora to estimate the rule probabilities. In this sense, we use a "true", not hypothesized, distribution based on a parallel corpus that maximizes the likelihood of the training data.

We will begin by briefly introducing our baseline formal syntax-based system in Sec. 2, then describe our proposed EM-SCFG algorithm in Sec. 3, followed by the experimental results in Sec. 4 and the conclusions in Sec. 5.

2. FORMAL SYNTAX-BASED BASELINE

The baseline system we used in this paper is a state-of-the-art formal syntax-based translation system, as described in detail in [2].

2.1. Model

All rules in $\mathcal{R}$ are paired with statistical parameters (i.e., weighted SCFG), which combines with other features to form our models using a log-linear framework. Our baseline model follows Chiang's hierarchical model [1] in conjunction with additional features, i.e., *abstraction penalty*. This makes our syntax-based model include a total of nine features.

Translation using SCFG for an input sentence $\mathbf{f}$ is casted as to find the optimal derivation on the source and target sides (as the grammar is synchronous, the derivations on source and target sides are identical). By "optimal", it indicates that the derivation D maximizes the following log-linear models over all possible derivations:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times \prod_i \prod_{X \to \langle \gamma, \alpha \rangle \in D} \phi_i(X \to \langle \gamma, \alpha \rangle)^{\lambda_i}, \quad (2)$$

where the set of $\phi_i(X \to \langle \gamma, \alpha \rangle)$ 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.

2.2. Decoder

The objective of our syntax-based decoder is to search for the optimal derivation tree D from a forest of trees that can represent the input sentence. The target side is mapped accordingly at each non-terminal node in the tree, and a traverse of these nodes obtains the target translation.

Our decoder implements a modified CKY parser in C++ with integrated n-gram language model scoring. During the search, chart cells are filled in a bottom-up fashion until a tree rooted from non-terminal is generated that covers the entire input sentence. The dynamic programming item we bookkeep is denoted as $[X, i, j; e_b]$, indicating a sub-tree rooted with X that has covered input from position i to j generating target translation with boundary words e_b. To speed up the decoding, a pruning scheme similar to the cube pruning [1] is performed during search.

3. ALGORITHMS

3.1. EM Algorithm for PCFG

In this section, we first briefly introduce the EM algorithm for probabilistic context-free grammar (PCFG) for monolingual corpus, following the formulism in [5]. An PCFG is defined as a 4-tuple (V_N, V_T, S, R), where V_N, V_T, and R are a set of non-terminals, terminals and rules respectively, and S is the start symbol. For each rule r in the rule set R, there is a statistical parameter $P(r)$ ($r \in R$) associated with it. The EM algorithm can be used to iteratively estimate the parameters $P(r)$ for PCFGs [6]. Given a training corpus C of terminal symbol sequences, the 'E' step of EM algorithm calculates all the possible derivations (parses) Φ for every terminal sequence $\mathbf{w} = w_1^l = w_1, w_2, \ldots, w_l$ in C, and then uses that information to calculate an expected count $c(r)$ that each rule r was used to produce the corpus:

$$c(r) = \sum_{\mathbf{w} \in C} \sum_{\phi \in \Phi} P(r, \phi | \mathbf{w}) \quad (3)$$

where $P(r, \phi|\mathbf{w})$ is the probability of rule r in the derivation ϕ given the terminal sequence $\mathbf{w}$. The 'M' step of EM algorithm then uses the expected counts to update the rule probabilities $P(r)$, by simply setting each rule probability $P(r)$ to expected count $c(r)$ and then normalizing so that the probability constraints are satisfied:

$$P(r) = \frac{c(r)}{\sum_{r' \in R: L(r') = L(r)} c(r')} \quad (4)$$

where $L(r)$ denotes the left-hand side of rule r.

The calculation of expected counts in the 'E' step requires a computational complexity of $O(|\mathbf{w}|^3)$. The inside-outside algorithm utilizes the dynamic programming to efficiently calculate the expected counts, by finding all derivations for a terminal sequence at the same time. It defines two conditional probabilities for a non-terminal $A \in V_N$ – the inside probability $\beta_{i,j}(A)$, and the outside probability $\alpha_{i,j}(A)$:

$$\beta_{i,j}(A) = P(A \stackrel{*}{\Rightarrow} w_i^j) \quad (5)$$

$$\alpha_{i,j}(A) = P(S \stackrel{*}{\Rightarrow} w_1^{i-1} A w_{j+1}^l) \quad (6)$$

If we assume that the PCFG grammar is in Chomsky normal form, then each rule r takes the form of either $A \to BC$ or $A \to a$ ($A, B, C \in V_N, a \in V_T$). The expected counts $c(r|w_1^l)$ contributed by the terminal sequence $\mathbf{w}$ can be efficiently calculated via the inside-outside algorithm as follows:

$$c(A \to BC | w_1^l) = \frac{P(A \to BC)}{P(w_1^l)} \times \sum_{n=1}^{l-1} \sum_{i=1}^{l-n} \sum_{j=1}^{n} \alpha_{i,i+n}(A) \beta_{i,i+j-1}(B) \beta_{i+j,i+n}(C)$$

$$c(A \to a | w_1^l) = \frac{P(A \to a)}{P(w_1^l)} \sum_{n=1}^{l} \alpha_{n,n}(A) \quad (7)$$

With the expected counts, the rule probability $P(r)$ can be iteratively updated according to Eq. 4 during the EM iterations.

3.2. EM Algorithm for SCFG

An SCFG defines the rewriting relationship between two sides – the source and the target languages. There are therefore some differences between the EM algorithm for PCFGs

and that for SCFGs. First, let $\mathbf{f} = f_1^M = f_1, f_2, \ldots, f_M$ and $\mathbf{e} = e_1^N = e_1, e_2, \ldots, e_N$ be the source and the target terminal sequences respectively. For SCFGs we need to consider the terminal sequences from both the source side $\mathbf{f}$ and the target side $\mathbf{e}$ at the same time, which is a two-dimensional constrain. Second, since there is only one non-terminal X, which is also the start symbol, in our SCFG translation model, we can safely omit the non-terminal parameter in the definition of the inside and outside probabilities. Consequently, we define the following inside and outside probabilities for SCFGs, with X explicitly denoted:

$$\beta_{i,j;k,l}(X) = P(X \stackrel{*}{\Rightarrow} (f_i^j; e_k^l)) \quad (8)$$

$$\alpha_{i,j;k,l}(X) = P(X \stackrel{*}{\Rightarrow} (f_1^{i-1} X f_{j+1}^M; e_1^{k-1} X e_{l+1}^N)) \quad (9)$$

where $(f_i^j; e_k^l)$ denotes the parallel terminal sequences for the source and the target sides.

The inside-outside algorithm for PCFGs can be similarly applied here for SCFGs to efficiently calculate the inside and the outside probabilities. The most expensive part to compute within the inside-outside algorithm is the *chart parsing* — to form the bilingual chart for parallel sentences $\mathbf{f}$ and $\mathbf{e}$. We use a variant of CKY algorithm in the same manner as Chiang [1] to construct the bilingual chart by synchronously parsing the source and the target sentences over two dimensions, with a complexity of $O(|\mathbf{e}|^3|\mathbf{f}|^3)$. More specifically, we loop over all the terminal subsequences f_i^j of $\mathbf{f}$ and e_k^l of $\mathbf{e}$ synchronously. For each pair of subsequences $(f_i^j; e_k^l)$, we increasingly fill the chart cell located at $[i, j; k, l]$ in a bottom-up style, with all the following hypotheses obtained from the baseline translation model by: 1) phrasal rules $X \rightarrow \langle f_i^j, e_k^l \rangle$; 2) abstract rules with one non-terminal X derivable for $(f_i^j; e_k^l)$; 3) abstract rules with two non-terminals X_1 and X_2 derivable for $(f_i^j; e_k^l)$. If we can reach the cell located at $[1, M; 1, N]$, then this means the parallel sentences $(f_1^M; e_1^N)$ are parsable given the translation model. We simply discard those parallel sentences that are unreachable. After filling the bilingual chart, we can use dynamic programming to recursively calculate the inside and the outside probabilities efficiently. As noted above, we can omit the non-terminal parameter for the inside and outside probabilities, which means that each cell at $[i, j; k, l]$ in the bilingual chart is associated with an inside probability $\beta_{i,j;k,l}$ and an outside probability $\alpha_{i,j;k,l}$, which are parameterized by the location (coordinations) of cells.

Similarly, the expected count for each rule $r \in \mathcal{R}$ can be calculated based on the inside and the outside probabilities using one of the following equations, depending on the number of non-terminals in the rule. This is a generalization of the ITG alignment algorithm [7].

- r_0 – phrasal rules without non-terminals:

$$c(r_0) = \sum_{(f_1^M; e_1^N) \in \mathcal{C}} \frac{P(r_0)}{P((f_1^M; e_1^N))} \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_{m,m;n,n} \quad (10)$$

- r_1 – SCFG rules with one non-terminal X_1:

$$c(r_1) = \sum_{(f_1^M; e_1^N) \in \mathcal{C}} \frac{P(r_1)}{P((f_1^M; e_1^N))} \times \sum_{m=1}^{M} \sum_{i=1}^{M-m+1} \sum_{n=1}^{N} \sum_{k=1}^{N-n+1} \alpha_{i,i+m;k,k+n} \beta_{[X_1]} \quad (11)$$

- r_2 – SCFG rules with two non-terminals X_1 and X_2:

$$c(r_2) = \sum_{(f_1^M; e_1^N) \in \mathcal{C}} \frac{P(r_2)}{P((f_1^M; e_1^N))} \times \sum_{m=1}^{M} \sum_{i=1}^{M-m+1} \sum_{n=1}^{N} \sum_{k=1}^{N-n+1} \alpha_{i,i+m;k,k+n} \beta_{[X_1]} \beta_{[X_2]} \quad (12)$$

where $\mathcal{C} = \{(f_1^M; e_1^N)\}$ is the parallel corpus, $P((f_1^M; e_1^N))$ equals to the inside probability of cell $[1, M; 1, N]$ in the bilingual chart, and $\beta_{[X_i]}$ represents the inside probability of the chart cell pointed by the subphrase X_i. Since the phrasal rule probabilities r_0 are reasonably estimated based on the occurrence statistics of parallel corpus, we only update the probabilities for abstract rules in this work, that is, we leave the phrasal rules r_0 unchanged.

We can re-estimate $P(\gamma|\alpha)$ or $P(\alpha|\gamma)$ based on the expected counts using different normalization denominator:

$$P^*(\gamma|\alpha) = \frac{c_{\gamma|\alpha}(X \rightarrow \langle \gamma, \alpha \rangle)}{\sum c_{\gamma|\alpha}(X \rightarrow \langle *, \alpha \rangle)} \quad (13)$$

$$P^*(\alpha|\gamma) = \frac{c_{\alpha|\gamma}(X \rightarrow \langle \gamma, \alpha \rangle)}{\sum c_{\alpha|\gamma}(X \rightarrow \langle \gamma, * \rangle)} \quad (14)$$

In summary, the training procedure to re-estimate the rule parameters of a formal syntax-based translation model is the following:

1. Read in the baseline formal syntax-based translation model, which has four features: $P(\gamma|\alpha)$,$P(\alpha|\gamma)$, $P_w(\gamma|\alpha)$,and $P_w(\alpha|\gamma)$.
2. For each pair $(\mathbf{f}; \mathbf{e})$ in the training parallel corpus $\mathcal{C}$:
   (a) Construct the bilingual chart for parsing;
   (b) Initialize the rule probabilities with $P(r) = P(\gamma|\alpha)$, calculate the inside and the outside probabilities for chart cells, and accumulate the expected count $c_{\gamma|\alpha}(r)$ for each rule $r \in \mathcal{R}$;
   (c) Initialize the rule probabilities with $P(r) = P(\alpha|\gamma)$, calculate the inside and the outside probabilities for chart cells, and accumulate the expected count $c_{\alpha|\gamma}(r)$ for each rule $r \in \mathcal{R}$;
3. Update $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$ according to Eq.13-14, and save the updated parameters $P^*(\gamma|\alpha)$ and $P^*(\alpha|\gamma)$.
4. Repeat steps 2-3 for several iterations.

4. EXPERIMENTS AND RESULTS

We evaluate our algorithm on the IWSLT 2006 Chinese to English translation task, using a different segmentation on the Chinese side than the original IWSLT 2006 data. The statistics for this data are shown in Table 1. The vocabulary size is 9,812 for English and 11,145 for Chinese after segmentation. There is no case and punctuation information in the data.

Table 1. Data statistics for IWSLT 2006 training, dev, and test sets, *using the first of multiple reference translations only.

Dataset		#Sentence	#Words
Training	Chinese	39,953	295,570
	English	39,953	306,378
Dev set	Chinese	489	5,214
(dev4)	English	489	$5,356^*$
Test set	Chinese	500	5,550
	English	500	$6,254^*$

We compare the updated models with the baseline model as described in Sec. 2. For the baseline syntax-based system, we generated a total of $1,002,436$ rules and used 9 features. The baseline translation model was updated according to the algorithms describe in Sec. 3. We trained a 4-gram language model for English using the English side of the parallel corpus. Minimum-error-rate(MET) training [8] was conducted on the dev4 set to optimize feature weights maximizing the BLEU scores up to 4-grams, and the obtained feature weights were blindly applied on the test set. We use parallel computing for the EM training, making it scalable to larger corpora. Totally 5% of non-parsable parallel sentences are skipped.

Table 2 shows the performance of the proposed algorithm evaluated on the dev4 and test sets using the BLEU metric with 7 references. We can see that the BLEU scores on both dev4 and test sets generally increase with more and more EM iterations, peaking at the 10th iteration. After that, the BLEU scores decrease, due to some overfitting effects. The best BLEU scores obtained by the proposed EM-SCFG algorithm at the 10th iteration are 1.0% and 1.3% absolutely better than the baseline model on the dev4 and test sets respectively. A significant testing demonstrates that the improvement of the BLEU score on the test set is significant, with $p < 0.03$.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce an EM algorithm for SCFG in a formal syntax-based translation system. The proposed algorithm avoids the need to hypothesize an inaccurate distribution for rule parameter estimation. Experimental results on the IWSLT 2006 Chinese to English task show statistically significant improvements in BLEU scores.

One problem of the proposed EM-SCFG algorithm is the overfitting. It is not easy to determine the convergence of EM training to avoid the overfitting. In the future, we will continue our research in this direction towards the better estimation of rule parameters, for example, approaches beyond maximum likelihood estimation such as variational Bayes, or a full Bayesian framework.

Table 2. The BLEU score (%) results on the IWSLT 2006 Chinese-to-English task.

Models		DEV4	TEST
Baseline		20.31	21.24
EM-SCFG	Iteration-1	20.60	21.06
	Iteration-2	20.66	21.56
	Iteration-3	20.91	21.43
	Iteration-4	20.93	22.01
	Iteration-5	20.47	21.49
	Iteration-6	21.11	21.68
	Iteration-7	21.06	21.86
	Iteration-8	20.36	20.76
	Iteration-9	21.16	21.36
	Iteration-10	**21.30**	**22.57**
	Iteration-11	21.24	22.09

6. REFERENCES

[1] David Chiang, "Hierarchical phrase-based translation," *Comput. Linguist.*, vol. 33, no. 2, pp. 201–228, 2007.

[2] Bowen Zhou, Bing Xiang, Xiaodan Zhu, and Yuqing Gao, "Prior derivation models for formally syntax-based translation using linguistically syntactic parsing and tree kernels," in *Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, 2008.

[3] Franz Och and Hermann Ney, "Improved statistical alignment models," in *Proc. of ACL*, 2000, pp. 440–447.

[4] Franz Och and Hermann Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[5] Kenichi Kurihara and Taisuke Sato, "An application of the variational bayesian approach to probabilistic context-free grammars," in *IJCNLP-04 Workshop Beyond Shallow Analyses*, 2004.

[6] James K. Baker, "Trainable grammars for speech recognition," in *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, 1979.

[7] Dekai Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, 1997.

[8] F. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL*, 2003, pp. 160–167.