*2012 International Conference on Network and Computational Intelligence (ICNCI 2012)*
***IPCSIT vol. 46 (2012) © (2012) IACSIT Press, Singapore***

# Feature-Based ITG for Unsupervised Word Alignment

Zhonghua Li[1+], Jun Lang[2], Yun Huang[3] and Jiajun Chen [1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Institute for Infocomm Research, Singapore
[3]Department of Computer Science, School of Computing, National University of Singapore

**Abstract.**Inversiontransduction grammar (ITG) [1]is an effective constrainttoword alignment search space. However, the traditional unsupervised ITG word alignment model is incapable of utilizing rich features. In this paper, we propose a novel feature-based unsupervised ITG word alignment model. With the help of rich features and $L_1$ regularization, a compact grammar is learned. Experiments on both word alignment and end-to-end statistical machine translation (SMT) task show that our model achieves better performance than the traditional ITG model with the AER of word alignment improved by 3 points and the BLEU score of SMT improved by 0.8 points.

**Keywords:**Statistical machine translation; Unsupervised word alignment; Feature;$L_1$regularization

## 1. Introduction

The goal of machine translation is to translate a text given in some source language into a target language. We are given a source string $f_1^J = f_1 \cdots f_j \cdots f_J$ of length *J*, which is to be translated into a target string $e_1^I = e_1 \cdots e_i \cdots e_I$ of length*I*. Statistical machine translation can be formulated as follows:

$$e^* = argmax_e P(e|f) = argmax_e P(f|e)P(e) \quad (1)$$

where $e$ is the target sentence, and $f$ is the source sentence. $P(e)$isthe target language model and $P(f|e)$ is the translation model.

Word alignment is a crucial step in the training of most statistical machine translation (SMT) systems. It is the task of inference word correspondences between bilingual sentence pairs. Fig.1 shows an example of word alignment between a Chinese sentence and an English sentence. The Chinese and English words are listed horizontally and vertically, respectively.We use the shading boxes to indicate the correspondences between the words in the two languages. The “China” is aligned to “zhongguo”; “development” is aligned to “fazhan”; “economy” is aligned to “jingji”; “‘s ”is aligned to “de”.

The advantage of discriminative models is that they can use rich features, whereas, the advantage of generative models is that they do not need manual aligned data. In this paper, we combine the advantages of these two learning paradigms and propose a feature-based unsupervised word alignment model, which can use rich features, but without relying on manual aligned data. We use the inversion transduction grammars (ITG) [1] as the alignment search space constraint. ITG has been widely used for word alignment. Especially the supervised ITG models [7, 8, 9], one reason for their good performance is that they can incorporate rich features, which are knowledge source and the key to achieve better performance. However, all these models are supervised learning-based, and thus need the manual aligned training data, which are only available for limited language pairs and expensive to create. Moreover, as pointed by [10] manual alignments impose a

[+]Corresponding author. Tel.: +(86) 15951986898
*E-mail address*:(balance.lzh@gmail.com).

commitment to a particular preprocessing regime. Unsupervised ITG models [11, 12], have also been extensively studied, but they are incapable of utilizing features.To the best of our knowledge, our model is the firstfeature-based unsupervised ITG word alignment model.Moreover, in order to handle high-dimensional feature space, $L_1$regularization is used in the objective function, which leads a compact model to be learned.This is an alternativeapproach to achieve a sparse solution compared with the previous work which usually usesa sparse prior in Bayesian formulation [12, 13].

The rest of this paper is organized as follows. Section 2 describes the traditional ITG model and our new feature-based ITG model. Section 3 describes the learningalgorithm of our model. Section 4 describes the features which are used in our model. Section 5presents some experimental results and related results analysis. Section 6 reviews some related works and presents our concluding remarks and future works.

# 2. Models

## 2.1. Inversion Transduction Grammars

ITG [1] is a well-studied synchronous grammar formalism in which the source and target sentence are generated simultaneously. The right hand side of the ITG rules is either two non-terminals or a terminal sequence. In our work, we adopt the left heavy grammar [1]:

S → A | B | C

A → [A B] | [B B] | [C B] | [A C] | [B C] | [C C]

B →<A A> | <B A> | <C A> | <A C> | <B C> | <C C>

C → e/f | ϵ/f | e/ϵ

Rules with non-terminals inside square brackets and angled brackets are called straight rules and inverted rules, respectively. The straight rules expand the left hand side into two symbols in the same order in the two languages. The inverted rules expand the left hand side into two symbols in inverted order in the two languages. Fig.2 shows an example of word alignment and ITG parse tree for a bilingual sentence. The "S","A","C" in the tree are the non-terminals of the grammar.

## 2.2. Feature-Based ITG

Inspired by [14], we change the parameter form of the traditional ITG and model the probability of terminal rules in a log-linear form:

$$P\left(r_c|\overrightarrow{\lambda}\right) = \frac{exp<\overrightarrow{\lambda}, \vec{f}(r_c)>}{\sum_{r_c' \in R_c} exp<\overrightarrow{\lambda}, \vec{f}(r_c')>} \quad (2)$$

Where$r_c$ is the terminal rule, $R_c$is the whole terminal rules of the grammar,$\vec{f}(r_c)$is the feature vector of $r_c$, $\overrightarrow{\lambda}$ is the weight vector, and <> represents the inner product.

For non-terminal rules, their original forms are kept unchanged. The feature-based ITG model is abbreviated as F-ITG in the following sections. Because a log-linear sub-model is adopted, there are two advantages of the F-ITG model. First, rich features can be incorporated into the rules.Second, feature selection technology can be used to learn a compact grammar.

We use the maximum likelihood estimation method to train the F-ITG model. The objective function of the model is the log-likelihood of the training data along with the $L_1$ regularization term.

$$L\left(\overrightarrow{\lambda}\right) = log\, P\left(X|\overrightarrow{\lambda}\right) - k\left\|\overrightarrow{\lambda}\right\|_1 \quad (3)$$

where$X$ denotes the bilingual corpus and$k$ is the $L_1$ coefficient. The $L_1$ regularization is used to control the model complexity and acts as a feature selection mechanism [15].

# 3. Parameter Estimation

One advantage of the F-ITG model is that EM-style algorithm can still be used to estimate the model parameters.In the E-step, the inside-outside algorithm is used to collect the expected counts of rules

according to the current rule's probabilities. This step remains unchanged from traditional ITG model. Given the rule's expected counts, the M-step tunesparametersto maximize the regularized expected complete log likelihood[1].

$$l(\vec{\lambda}, \vec{e}_{r_c}) = \sum_{r_c \in R_c} e_{r_c} \log P(r_c|\vec{\lambda}) - k\|\vec{\lambda}\|_1 \quad (4)$$

where$e_{r_c}$istheexpected count of terminal rule calculated in the E-step.

Theobjective function in (4) is not everywhere differentiable due to the $L_1$ regularization.So we adopt the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) algorithm [16] to optimize it[2]. The OWL-QNalgorithmneeds to know the gradient of the first term in the objective function, which can be computed as:

$$\nabla_{\vec{\lambda}} l = \sum_{r_c \in R_c} e_{r_c} * \vec{\Delta} \quad (5)$$

where$\vec{\Delta} = \vec{f}(r_c) - \sum_{r_c' \in R_c} P(r_c'|\vec{\lambda})\, \vec{f}(r_c')$

The IBM model 1 is trained on the bilingual corpus and the translation probabilities are used to initialize the terminal rule's probabilities. The fixed-link pruning [17] methodis used in our system to speed up the training. The pseudo code of the overall training procedure is given in Algorithm 1.In the E-step (line 3), the expected counts$\vec{e}_{r_n}$for non-terminal rules and $\vec{e}_{r_c}$for terminal rules are calculated. In the M-step, the non-terminal rule probabilities are estimated by normalizing expected counts (line 4-6), the OWL-QN is used to tune$\vec{\lambda}$ for terminal rules (line 7-11).

**Algorithm 1** Learning Algorithm for F-ITG

**Input**: Bilingual sentence pairs
**Output**: Probabilistic ITG

1 Initialize rule probabilities
2 **Repeat**
3 Compute $\vec{e}_{r_n}$ , $\vec{e}_{r_c}$
4 **for** $r_n \in R_n$**do**
5 $P(r_n) \propto e_{r_n}$
6 **end for**
7 **repeat**
8 Compute $l(\vec{\lambda}, \vec{e}_{r_c})$
9 Compute $\nabla_{\vec{\lambda}} l$
10 $\vec{\lambda} \leftarrow CLIMB(\vec{\lambda}, l, \nabla l)$
11 **until** convergence
12 Compute $P(r_c|\vec{\lambda})$
13 **until** convergence

# 4. Features

In this section, we describe the detailed features used in our model.

## 4.1. Basic Feature

To capture the co-occurrence of bilingual terminals, the basic feature is designed for each terminal rule.

$$f_{b:(\hat{e},\hat{f})}(e,f) = \begin{cases} 1, & if\ \mathrm{e} = \hat{\mathrm{e}} \text{ and } f = \hat{f} \\ 0, & otherwise \end{cases}$$

where $\hat{e}$ and $\hat{f}$are some specified source and target word, respectively. Note that if we only use the basic features and set weightsproperly (proportional to the logarithm of normalized relative frequency), thefeature-based ITG model is degenerated to traditional ITG model.

### 4.2. Dictionary Feature

Dictionaries are good resources for word alignment.In our experiments, the LDC Chinese-English Translation Lexicon (LDC2002L27) is used for defining the dictionary feature.

$$f_d(e,f) = \begin{cases} 1, & if\ entry\ (e,f) \text{ is in dictionary} \\ 0, & otherwise \end{cases}$$

### 4.3. Part-of-Speech Feature

Part-of-speech(POS)can provide some useful information for alignment. For example, noun word in one languageis often aligned to noun word in another language.Since the POS tag sets of English and Chinese are different, we have to map between the two sets.We first use the StanfordPOS tagger to tag both the Chinese side and the English side of whole corpus. Then, the GIZA++ [18] is adopted to train an intersection wordalignment on the training corpus[3]. Finally, the POS feature is defined as:

$$f_p(e,f) = \begin{cases} 1, & if\ P_w(e|f) * P_w(f|e) > \delta^2 \\ 0, & otherwise \end{cases}$$

where$P_w(e|f)$ and $P_w(f|e)$ are conditional probabilities between POStags, and the threshold$\delta^2$is set to 0.5 in experiments.

## 5. Experiments

In this section, the effectiveness of the proposed model is verified on the word alignment task and the end-to-end machine translation task.

### 5.1. Word Alignment Evaluation

As in previous work [7], the Chinese-English hand-aligned portion of the 2002 NIST MT evaluation set (WAAC for short) is adopted for evaluating word alignment performance. The Alignment Error Rate (AER) [19], precision and recall are reported. The definitions of them are as follows.

$$AER = 1 - (|A \cap P| + |A \cap S|)/(|A| + |S|)$$

$$Precision = |A \cap P|/|A|$$

$$Recall = |A \cap S|/|S|$$

where A is word alignments generated by systems, P and S are word alignments marked as “possible” and “sure” in the golden standard, respectively. Note that the lower of AER, the better of the model.

Our model is trained on WAAC (without word alignments) and FBIS (LDC2003E14) datasets. For FBIS corpus, sentence pairs with source/target length ratio more than 3 or less than 1/3 are removed out to reduce errors of wrongly aligned sentence pairs. To test the performance of different data size, we build a small dataset with length limit 15 and a large dataset with length limit 35 for both WAAC and FBIS corpora. Table 1 gives some statistics.

Table. 1:Corpus statistics

| Dataset | Source | Sent | Word[C/E] | Type[C/E] |
|---|---|---|---|---|
| small | WAAC | 97 | 906/1,064 | 564/577 |
| | FBIS | 23K | 214K/259K | 16K/13K |
| large | WAAC | 350 | 6.4K/7.5K | 2K/2K |
| | FBIS | 123K | 2.2M/2.8M | 40K/34K |

The performances of F-ITG and the baseline of traditional ITG are compared. We also report the alignments generated by GIZA++ for reference. For GIZA++, the default setting up to model 4 is used, and the “grow-diag-final-and” heuristic is used to obtain symmetric alignments. Table 2 shows the evaluation results. The proposed F-ITG model outperforms the traditional ITG significantly on both datasets for all three metrics. This is reasonable since additional resources could be easily incorporated as features and contribute to the overall performance. Compared to GIZA++, we observe that GIZA++ achievesbetter recall than F-ITG on both datasets. The reason should be that the “grow-diag-final-and” heuristicadds links that exist in theunion of the bi-directional alignments.While in F-ITG and ITG,theempty links are explicitly

modeled by theϵtoken,so the learned alignments often have more empty-linked words, resulting in low-recall but higher precision.

Table. 2:Word alignment result on small/large datasets

| Dataset | Model | Precision | Recall | AER |
|---|---|---|---|---|
| Small | ITG | 78.5 | 73.6 | 23.8 |
| | F-ITG | **85.4** | 74.5 | **20.4** |
| | GIZA++ | 59.9 | **74.9** | 33.7 |
| Large | ITG | 74.3 | 72.8 | 26.4 |
| | F-ITG | **78.1** | 74.2 | **23.9** |
| | GIZA++ | 71.1 | **80.2** | 24.9 |

Table.3:Grammar statistics

| Model | Total Rules | Active | Ratio |
|---|---|---|---|
| ITG | 8,614,343 | 606,304 | 7.04% |
| F-ITG | 8,614,343 | 82,959 | 0.96% |

As mentioned in Section 2, the $L_1$ regularization can act as a feature selection mechanism. Most feature weights (more than 98%) in our model are optimized to zero, which leads to a compact grammar. The rule's distribution of the F-ITG is more non-uniform than the ITG's. To compare the two grammars, the rule with probability larger than $1.0 * 10^{-10}$is defined as active rule. The size of active rules of F-ITG is far less than the traditional ITG as showed in Table 3. Fig. 3 gives a snip of our grammar. The left column is the probability of the rule. The right column is the rule itself. Note that, we use the index of the word to represent the word itself in the rule.

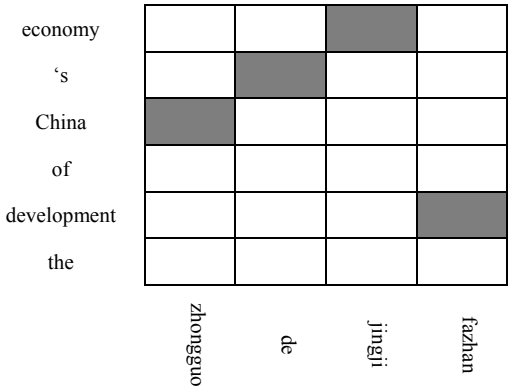
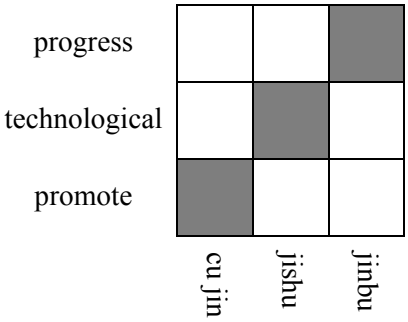



Fig. 1: Word Alignment Matrices



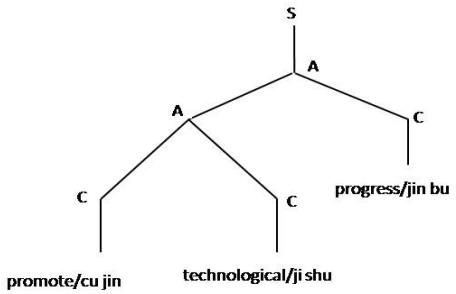



Fig. 2:Word Alignment Matrices and ITG Parse Tree

| | | | |
|---|---|---|---|
| 0.968316 | S → A | 0.133342 | B →< A C > |
| 0.0297967 | S → B | 0.133605 | B →< B C > |
| 0.001887 | S → C | 0.309371 | B →< C C > |
| 0.0920312 | A →[ A B ] | 1e-10 | C →1 \| 1863 |
| 0.00692773 | A →[ B B ] | 1e-10 | C →1 \| 551 |
| 0.0357791 | A →[ C B ] | 1e-10 | C →1 \| 172 |
| 0.617971 | A →[ A C ] | 1e-10 | C →1 \| 847 |
| 0.0568261 | A →[ B C ] | 1.31398e-05 | C →1 \| 295 |
| 0.190465 | A →[ C C ] | 1e-10 | C →1 \| 3579 |
| 0.14864 | B →<A A > | 1e-10 | C →1668 \| 1 |
| 0.102818 | B →< B A > | 4.58867e-05 | C →1668 \| 1863 |
| 0.172223 | B →< C A > | | |

Fig. 3:A snip of our learned grammar

## 5.2. SMT Experiments

We also carry out SMT experiments to check if our better word alignment leads to better SMT performance. The large dataset described above is used as training corpus. The NIST 2003 and NIST 2005 Chinese-to-English MT evaluation test sets are used as development set and test set, respectively. A 5-gram language model is trained on the Xinhua portion of the English Gigaword corpus (LDC2003T05) by the SRILM toolkit [20].Moses [21] is used as our decoder. The MERT [22] tuning procedure is run 10 times. The average BLEU [23] scores are reported. Table 4 presents the SMT results based on F-ITG, traditional ITG, and GIZA++. The F-ITG model obtains more than 0.8BLEU improvementsover the baseline and a slightly better BLEU score than GIZA++.

Table.4:SMT Experiments Result

| Model | Dev | Test |
|---|---|---|
| ITG | 27.06 | 26.42 |
| F-ITG | **27.76** | **27.24** |
| GIZA++ | 26.68 | 27.12 |

# 6. Related Work and Conclusion

The approach of using features in the unsupervised model is useful [10, 24, 25]. Our work is similar to [14], But they do not study the alignment space under the ITG constraints and do not explore the usage of$L_1$

regularization for sparsity. In order to learn a compact and simple solution, previous works usually usea sparse prior in Bayesian formulation [12, 13, 26, 27]. We use an alternative approach. Therule's probability is parameterized in a log-linear form anda $L_1$ regularization is used to achieve the same goal. A compact grammar is benefit to word alignment, which is also verified by [28]. They search for good alignment that minimized the size of the induced bilingual dictionary.

In this paper, a simple but effective feature-based unsupervised word alignment model under the ITG constraints is presented. With the help of the rich features and the $L_1$ regularization, a compact grammar is learned. Experiments on both word alignment and SMT show that, the F-ITG model can achieve significant improvementthan the traditional ITG model. In future, we will continue working on this line of research. In this work, there are three features are used in the model. We will design more features into the rules to achieve further improvement. The non-terminal rules stay unchanged in current model. We will incorporate features into the non-terminal rules.

## 7. Acknowledgements

This work was done during the first author's internship at Institute for Infocomm Research, Singapore. We would like to thank Xinyan Xiao and Shujian Huang for their helpful discussion regarding this work.

[1] The term about non-terminal rules are not included in (4). The probabilities of the non-terminals can be computed as in traditional ITG model.

[2] We use the open-source implementation of OWL-QN:http://www.chokkan.org/software/liblbfgs/

[3] This feature does not rely on GIZA++. We run GIZA++ only one time for getting a map between the two tag sets.

## 8. References

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

[2] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263–311.

[3] Stephan Vogel, Hermann Ney, and ChristophTillmann, "HMM-based word alignment in statistical translation," in Proceedings of the 16thInternational Conference on Computational Linguistics, 1996, vol. 2, pp. 836–841.

[4] Percy Liang, Ben Taskar, and Dan Klein, "Alignment by agreement," in Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, 2006, pp. 104–111.

[5] Ben Taskar, Lacoste-Julien Simon, and Klein Dan, "A discriminative matching approach to word alignment," in Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 2005, pp. 73–80.

[6] Yang Liu, Qun Liu, and Shouxun Lin, "Log-linear models for word alignment," in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), 2005, pp. 459–466.

[7] Aria Haghighi, John Blitzer, John DeNero, and Dan Klein, "Better word alignments with supervised ITG models," in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009, pp. 923–931.

[8] Shujie Liu, Chi-Ho Li, and Ming Zhou, "Discriminative pruning for discriminative ITG alignment," in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 316–324.

[9] Shujian Huang, Stephan Vogel, and Jiajun Chen, "Dealing with spurious ambiguity in learning ITG-based word alignment," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 379–383.

[10] Chris Dyer, Jonathan H. Clark, AlonLavie, and Noah A. Smith, "Unsupervised word alignment with arbitrary features," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 409–419.

[11] Hao Zhang and Daniel Gildea, “Stochastic lexicalized inversion transduction grammar for alignment,” in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), 2005, pp. 475–482.

[12] Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea, “Bayesian learning of non-compositional phrases with synchronous parsing,” in Proceedings of ACL-08: HLT, 2008, pp. 97–105.

[13] Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne, “A gibbs sampler for phrasal synchronous grammar induction,” in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009, pp. 782–790.

[14] Taylor Berg-Kirkpatrick, Alexandre Bouchard-Cote, John DeNero, and Dan Klein, “Painless unsupervised learning with features,” in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 582–590.

[15] Andrew Y. Ng, “Feature selection, l1 vs. l2 regularization, and rotational invariance,” in Proceedings of the twenty-first international conference on Machine learning, 2004, ICML ’04.

[16] Galen Andrew and JianfengGao, “Scalable training of l1-regularized log-linear models,” in Proceedings of the 24th international conference on Machine learning, 2007, pp. 33–40.

[17] Colin Cherry and Dekang Lin, “Inversion transduction grammar for joint phrasal translation modeling,” in Proceedings of SSST, NAACL-HLT 2007 / AMTA Work-shop on Syntax and Structure in Statistical Translation,2007, pp. 17–24.

[18] Franz Josef Och and Hermann Ney, “A systematic comparison of various statistical alignment models,” Computational Linguistics, vol. 29, no. 1, pp. 19–51, 2003.

[19] Franz Josef Och and Hermann Ney, “Improved statistical alignment models,” in Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, 2000, pp. 440–447.

[20] Andreas Stolcke, “SRILM – an extensible language modeling toolkit,” in Proceedings of the 7th International Conference on Spoken Language Processing, 2002, pp. 901–904.

[21] Philipp Koehn and Hieu Hoang, “Factored translation models,” in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 868–876.

[22] Franz Josef Och, “Minimum error rate training in statistical machine translation,” in Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 2003, pp. 160–167.

[23] Kishore Papineni, SalimRoukos, Todd Ward, and WeiJing Zhu, “Bleu: a method for automatic evaluation of machine translation,” in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[24] Noah A. Smith and Jason Eisner, “Contrastive estimation: Training log-linear models on unlabeled data,” in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), 2005, pp. 354–362.

[25] Aria Haghighi and Dan Klein, “Prototype-driven grammar induction,” in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006, pp. 881–888.

[26] Mark Johnson, Thomas Griffiths, and Sharon Goldwa-ter, “Bayesian inference for PCFGs via Markov chain Monte Carlo,” in Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceed-ings of the Main Conference, 2007, pp. 139–146.

[27] Trevor Cohn, Sharon Goldwater, and Phil Blunsom, “Inducing compact but accurate tree-substitution grammars,” in Proceedings of Human Language Technolo-gies: The 2009 Annual Conference of the North Amer-ican Chapter of the Association for Computational Linguistics, 2009, pp. 548–556.

[28] TugbaBodrumlu, Kevin Knight, and Sujith Ravi, “A new objective function for word alignment,” in Proceedings of the Workshop on Integer Linear Program-ming for Natural Language Processing, 2009, pp. 28–35