Segmenting vs. Chunking Rules: Unsupervised ITG Induction via Minimum Conditional Description Length

Markus SAERS and **Karteek ADDANKI** and **Dekai WU**
Human Language Technology Center
Dept. of Computer Science and Engineering
Hong Kong University of Science and Technology
{masaers|vskaddanki|dekai}@cs.ust.hk

Abstract

We present an unsupervised learning model that induces phrasal inversion transduction grammars by introducing a minimum *conditional description length* (CDL) principle to drive search over a space defined by two opposing extreme types of ITGs. Our approach attacks the difficulty of acquiring more complex longer rules when inducing inversion transduction grammars via unsupervised bottom-up chunking, by augmenting its model search with top-down segmentation that minimizes CDL, resulting in significant translation accuracy gains. Chunked rules tend to be relatively short; long rules are hard to learn through chunking, as the smaller parts of the long rules may not necessarily be good translations themselves. Our objective criterion is a conditional adaptation of the notion of description length, that is conditioned on a fixed preexisting model, in this case the initial chunked ITG. The notion of minimum CDL (MCDL) facilitates a novel strategy for avoiding the pitfalls of premature pruning in chunking approaches, by incrementally splitting an ITG with reference to a second ITG that conditions this search.

1 Introduction

We describe an unsupervised approach to inducing phrasal inversion transduction grammars or ITGs (Wu, 1997) that employs a new theoretically well-founded minimum **conditional description length** (CDL) objective to explicitly drive two opposing, extreme ITGs towards one single ITG. Given one ITG initially composed of short rules learned by bottom-up chunking of short atomic rules, our method augments it with rules that are learned through top-down segmentation of long rules initialized by memorizing the parallel corpus. This offers an opportunity to capture longer non-compositional translations as explicit biterminal rules, which is hard for search to discover solely via bottom-up chunking. Iterative bottom-up chunking relies on composing two good translations into a longer good translation, which assumes that the long rules learned in this way are compositional. In contrast, iteratively segmenting an existing good translation into shorter good translations does not depend on assumptions about whether the resulting shorter rules can be further decomposed. Empirically, augmenting the chunked ITG with rules learned via top-down segmentation helps translation quality. However, the maximum likelihood objective is inadequate for this purpose; instead, we introduce the **minimum conditional description length** (MCDL) objective to drive the search for phrasal rules simultaneously from the two opposing types of ITG constraints, both of which have individually been empirically demonstrated to match phrase reordering patterns across translations well. In so doing, we aim to also provide an obvious basis for generalization to abstract translation schemas.

The necessity of MCDL as an alternative learning objective to standard maximum likelihood (ML) arises because the top-down rule segmentation search starts in a state where likelihood is already maximized, unlike bottom-up learning which can be driven with ML. The top-down search starts with all sentence pairs in the training corpus as biterminals, which maximizes the likelihood of the training data, but is guaranteed to generalize poorly to unseen data. There is no segmentation we can make to this grammar that would increase the likelihood of the training data, but we do nonetheless want to segment the existing rules so that the grammar has a chance to cover unseen

Proceedings of Recent Advances in Natural Language Processing, pages 584–592,
Hissar, Bulgaria, 7-13 September 2013.

data. The solution is to move away from pure ML; in this paper we will use minimum conditional description length, which has the likelihood of the training data as one component, but balances it with a notion of model size. MCDL allows us to make the training data less likely provided that the size of the grammar becomes smaller. Since the initial state of the top-down search has all the sentence pairs in the training data explicitly stored as biterminals, there is ample opportunity for shrinking the size of the grammar by segmenting the existing rules into reusable segments, and MCDL helps deciding when this is a good idea and when not. The difference between MCDL and minimum description length is that the lengths are subject to an external model. In our case, the external model is the bottom-up chunked ITG, which means that the auxiliary ITG being induced is tailored specifically towards augmenting it.

We choose to work with the well-defined and theoretically sound formalism of ITGs rather than over-engineered direct translation models (Koehn *et al.*, 2003) or feature-heavy transduction grammars (Chiang, 2005). The reason for this is twofold: (a) they allow for manual inspection, and (b) the assumptions stay the same through learning and testing. Being able to inspect the learned model is crucial for error analysis, but inspecting a typical state-of-the-art translation system is prohibitively hard. Phrasal direct translation systems rely heavily on the language model to compensate for the mistakes they make, as well as relying on a fine-tuned log-linear combination of several features to choose which lexical units to use. Pinning down exactly where and why an error occurred in this setup is very hard. The transduction grammar based approach is better in this respect, but the state-of-the-art typically relies on massive amounts, tens of thousands (Chiang *et al.*, 2009), of features. As a community, we still have no clear idea of why these features help translation, only that they do when the whole system pipeline is treated as a black box, but treating the system as a black box prevents effective error analysis. The state-of-the-art systems also relies on long and complicated learning pipelines that form ad-hoc models of how translation happens. These ad-hoc models differ significantly from the models of how translation happens that are used during actual translation, which violates the basic machine learning assumption that the same model should be used during training and testing. In contrast, the only difference between biparsing with ITGs (training) and decoding (testing) is that both sentences are given during biparsing, but only the input sentence during decoding—the model itself does not change, only the way it is used.

The space of possible ITG structures is intractably large, and there have been many attempts to introduce external constraints to guide the search. We do completely unsupervised search without introducing such constraints, which limits the scope of error analysis to the search strategy. Popular external constraints include word alignments (Chiang, 2005) and parse trees.

Word alignments are typically learned as a many-to-one function from one language into the other language (Brown *et al.*, 1993; Vogel *et al.*, 1996), but since no translation systems in use today actually rely on generating one output token at a time from zero or more input tokens, two opposing such functions are typically combined heuristically to form a many-to-many function between the input and output tokens. This is problematic, as it turns the alignments into hard constraints that are external to any model learned from them. Ironically, whenever transduction grammars are used to learn alignments these alignments are also treated as hard external constraints to the translation models that are learned from them (Cherry and Lin, 2007; Zhang *et al.*, 2008; Blunsom *et al.*, 2008, 2009; Haghighi *et al.*, 2009; Saers and Wu, 2009, 2011; Blunsom and Cohn, 2010; Burkett *et al.*, 2010; Riesa and Marcu, 2010; Saers *et al.*, 2010; Neubig *et al.*, 2011, 2012).

When parse trees are used to constrain the search they can be found on the input side only, making the resulting system a tree-to-string system, on the output side only, making it a string-to-tree system, or on both sides, making it a tree-to-tree system (Galley *et al.*, 2006). The grammarians who constructed the treebank—or the parser that it was created with, or the treebank that was used to train the parser—can and should not be expected to take into account the relationship between the language they are working with and all other languages on the planet, so the parse trees themselves run a real risk of matching the translation problem poorly.

We structure the paper so that we start by introducing conditional description length, which we will use to replace description length as the driving metric for the top-down rule-segmenting

ITG induction (Section 2). We then describe how we encode ITGs to measure their length in bits, which is a necessary component of any metric related to description length (Section 3). These two sections are the theoretical fundamental that we build the algorithms around. The first algorithm we describe is the baseline: top-down rule-segmenting ITG induction driven by minimum description length (Section 4). Although it is background, please bear with us as it serves an important role in contrasting conditional with unconditional, plain description length. This lays the ground work for the experimental contribution of the paper: Section 5 describes how we initialize an ITG by bottom-up rule-chunking, which is then augmented (Section 6) with rules learned through top-down rule segmentation as described in our second algorithm. This algorithm differs from the first in that it minimizes *conditional* description length rather than plain description length. We also test our model empirically in an experiment described in Section 7 and analyzed in Section 8. Finally, we offer some concluding remarks (Section 9).

2 Conditional Description Length

Conditional description length (CDL) is a general method for evaluating a model and a dataset given a preexisting model. This makes it ideal for augmenting an existing model with a variant model of the same family. In this paper we will apply this to augment an existing inversion transduction grammar (ITG) with rules that are found with a different search strategy. CDL is similar to description length (Solomonoff, 1959; Rissanen, 1983), but the length calculations are subject to additional constraints. When minimum CDL (MCDL) is used as a learning objective, all the desired properties of minimum description length (MDL) are retained: the model is allowed to become less certain about the data provided that it shrinks sufficiently to compensate for the loss in precision. MDL is a good way to prevent over-fitting, and MCDL retains this property, but for the task of inducing a model specifically to augment an existing model. Formally, CDL is:

$$DL\left(\Phi, D|\Psi\right) = DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right)$$

where Ψ is the fixed preexisting model, Φ is the model being induced, and D is the data. The total unconditional length is :

$$\begin{aligned} &DL\left(\Psi, \Phi, D\right) \\ &\quad = DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right) + DL\left(\Psi\right) \end{aligned}$$

In minimizing CDL, we fix Ψ instead of allowing it to vary as we would in full MDL; to be precise, we seek:

$$\begin{aligned} &\operatorname*{argmin}_{\Phi} DL\left(\Psi, \Phi, D\right) \\ &= \operatorname*{argmin}_{\Phi} DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right) + DL\left(\Psi\right) \\ &= \operatorname*{argmin}_{\Phi} DL\left(\Phi, D|\Psi\right) \\ &= \operatorname*{argmin}_{\Phi} DL\left(D|\Phi, \Psi\right) + DL\left(\Phi|\Psi\right) \end{aligned}$$

To measure the CDL of the data, we turn to information theory to count the number of bits needed to encode the data given the two models under an optimal encoding (Shannon, 1948), which gives:

$$DL\left(D|\Phi, \Psi\right) = -\lg P\left(D|\Phi, \Psi\right)$$

The CDL of the model is not necessarily expressible as a probability, and in this paper we will measure its length as the number of bits required to encode the model using a theoretical encoding.

To determine whether a model Φ has a shorter conditional description length, than another model Φ', it is sufficient to be able to subtract one length from the other. For the model length, this is trivial as we merely have to calculate the length of the difference between the two models in our theoretical encoding. For data length, we need to solve:

$$\begin{aligned} &DL\left(D|\Phi', \Psi\right) - DL\left(D|\Phi, \Psi\right) \\ &\quad = -\lg P\left(D|\Phi', \Psi\right) - -\lg P\left(D|\Phi, \Psi\right) \\ &\quad = -\lg \frac{P\left(D|\Phi', \Psi\right)}{P\left(D|\Phi, \Psi\right)} \end{aligned}$$

3 Encoding ITGs

By encoding an ITG, we turn the relatively complex data structure into a series of symbols—a message, whose length can be measured in bits. This section describes how we device this encoding scheme. An ITG consists of a set of nonterminal symbols, a set of L_0 symbols, a set of L_1 symbols, a set of rules and a start symbol. We notice that the only significance of the sets of nonterminal, L_0 and L_1 symbols is to categorize the symbols that occur in the rules, and the identity of the

start symbol constitutes a per-grammar constant. To measure the length of a grammar it is thus sufficient to measure and sum the lengths of all rules. We will measure the length by encoding the rule set as a sequence of symbols. We need one symbol for each of the nonterminal, L_0 and L_1 symbols of the ITG, as well as a meta symbol to separate rules and determine whether they are straight or inverted (unary rules are assumed to be straight). For conditional description length, rules that are found in Ψ can be excluded when measuring the length of Φ. Consider the following toy ITG:

$$\begin{array}{lll} S \rightarrow A, & A \rightarrow \langle AA \rangle, & A \rightarrow [AA], \\ A \rightarrow \text{have}/\text{有}, & A \rightarrow \text{yes}/\text{有}, & A \rightarrow \text{yes}/\text{是} \end{array}$$

which is conditioned on the following ITG:

$$\begin{array}{lll} S \rightarrow A, & A \rightarrow \langle AA \rangle, & A \rightarrow [AA], \\ A \rightarrow \cdots, & \cdots & \end{array}$$

Its serialized form would be:

$$[]A\text{have}\text{有}[]A\text{yes}\text{有}[]A\text{yes}\text{是}$$

Assuming a uniform distribution over the symbols, each symbol will require $-\lg\frac{1}{N}$ bits to encode (where N is the number of different symbols in the ITG). The above toy ITG has 8 symbols, meaning that each symbol requires 3 bits. The encoded message is 12 symbols long, making the ITG 36 bits long.

4 Baseline ITG

The natural baseline to compare ITGs learned by minimizing *conditional* description length is ITGs learned by minimizing *unconditional* description length, which we will describe in this section. This is the same model as described in Saers *et al.* (2013), which is repeated here to highlight the minimum changes needed to switch the objective function from minimum description length to minimum conditional description length.

The ITG is initialized with all sentence pairs as biterminals:

$$\begin{array}{rcl} S & \rightarrow & A \\ A & \rightarrow & e_{0..T_0}/f_{0..V_0} \\ A & \rightarrow & e_{0..T_1}/f_{0..V_1} \\ & \cdots & \\ A & \rightarrow & e_{0..T_N}/f_{0..V_N} \end{array}$$

where S is the start symbol, A is the nonterminal, N is the number of sentence pairs, T_i is the length of the i^{th} output sentence (making $e_{0..T_i}$ the i^{th} output sentence), and V_i is the length of the i^{th} input sentence (making $f_{0..V_i}$ the i^{th} input sentence). After the ITG has been initialized, its preterminal rules are iteratively segmented until no segmentations can be found that would shorten its description length. The parameters of the model is initialized as relative frequency of the sentence pairs/biterminals.

The segmentation algorithm relies on identifying parts of existing biterminals that could be validly used in isolation, and allow them to combine with other segments. We do this by proposing a number of sets of biterminal rules and a place to segment them, evaluate how the description length would change if we were to apply one of these sets of segmentations to the grammar, and commit to the best set. That is: we do a greedy search over the power set of possible segmentations of the rule set. The key component in the approach is the ability to evaluate how the description length would change if a specific segmentation was made in the grammar. This can be extended to a set of segmentations, which only leaves the problem of generating suitable sets of segmentations.

The key to a successful segmentation is to maximize the potential for reuse, either by being able to identify a segment across multiple rules. Consider the terminal rule:

$$\begin{array}{rcl} A & \rightarrow & \text{five thousand yen is my limit}/ \\ & & \text{我最多出五千日元} \end{array}$$

(Chinese romanization: wǒ zùi dūo chū wǔ qīan rì yúan). This rule can be split into three rules:

$$\begin{array}{rcl} A & \rightarrow & \langle AA \rangle, \\ A & \rightarrow & \text{five thousand yen}/\text{五千日元}, \\ A & \rightarrow & \text{is my limit}/\text{我最多出} \end{array}$$

Note that the original rule consists of 16 symbols (in our encoding scheme), whereas the new three rules consists of $4 + 9 + 9 = 22$ symbols. The bracketing inverted rule is likely to already be in the ITG, but the lexical rules still contain 18 symbols, which is decidedly longer than 16 symbols—and we need to get the length to be shorter if we want to see a net gain, since the length of the data is likely to be longer with the segmented rules. What we really need to do is find a way to reuse the lexical rules that came out of the segmentation. Now

suppose the ITG also contained this terminal rule:

$$A \rightarrow \text{the total fare is five thousand yen/总共的费用是五千日元}$$

(Chinese romanization: zǒng gòng de fèi yòng shì wǔ qīan rì yúan). This rule can also be split into three rules:

$$A \rightarrow [AA],$$
$$A \rightarrow \text{the total fare is/总共的费用是,}$$
$$A \rightarrow \text{five thousand yen/五千日元}$$

Again, the structural rule is likely to already be present in the ITG, the old rule was 19 symbols long, and the two new terminal rules are $12 + 9 = 21$ symbols long. Again we are out of luck, as the new rules are longer than the old one, and three rules are likely to be less probable than one rule during parsing. The way to make this work is to realize that the two existing rules share a bilingual affix—a **biaffix**: five thousand dollars translating into 五千日元. If we make the two changes at the same time, we get rid of $16 + 19 = 35$ symbols worth of rules, and introduce a mere $9 + 9 + 12 = 30$ symbols worth of rules. Making these two changes at the same time is essential, as the length of the five saved symbols can be used to offset the likely increase in the length of the data. And of course: the more rules we can find with shared biaffixes, the more likely we are to find a good set of segmentations.

The top-down search algorithm takes advantage of the above observation by focusing on the biaffixes found in the training data. Each biaffix defines a set of lexical rules paired up with a possible segmentation. We evaluate the biaffixes by estimating the change in description length associated with committing to all the segmentations defined by a biaffix. This allows us to find the best set of segmentations, but rather than committing only to the one best set of segmentations, we will collect all sets which would improve description length, and try to commit to as many of them as possible. The pseudocode can be found in Algorithm 1. It uses the methods `collect_biaffixes`, `eval_dl`, `sort_by_delta` and `make_segmentations`. These methods collects all the biaffixes in an ITG, evaluate the difference in description length, sorts candidates by these differences, and commits to a given set of candidates, respectively. To evaluate the DL of a proposed set of candidate segmentations, we need to calculate the difference in DL between the current model, and the model that would result from committing to the candidate segmentations:

$$\begin{aligned} &DL\left(\Phi', D\right) - DL\left(\Phi, D\right) \\ &= DL\left(D|\Phi'\right) - DL\left(D|\Phi\right) \\ &\quad + DL\left(\Phi'\right) - DL\left(\Phi\right) \end{aligned}$$

The model lengths are trivial, as we merely have to encode the rules that are removed and inserted according to our encoding scheme and plug in the summed lengths in the above equation. This leaves the length of the data, which is:

$$DL\left(D|\Phi'\right) - DL\left(D|\Phi\right) = -\lg \frac{P\left(D|\theta'\right)}{P\left(D|\theta\right)}$$

where θ and θ' are the parameters of Φ and Φ' respectively. This lets us determine the probability through biparsing with the model being induced. Biparsing is, however, a very expensive operation, and we are making relatively small changes to the ITG, so we will further assume that we can estimate the DL difference in closed form based on the model parameters. Given that we are splitting the rule r_0 into the three rules r_1, r_2 and r_3, and that the probability mass of r_0 is distributed uniformly over the new rules, the new grammar parameters θ' will be identical to θ, except that:

$$\begin{aligned} \theta'_{r_0} &= 0 \\ \theta'_{r_1} &= \theta_{r_1} + \frac{1}{3}\theta_{r_0} \\ \theta'_{r_2} &= \theta_{r_2} + \frac{1}{3}\theta_{r_0} \\ \theta'_{r_3} &= \theta_{r_3} + \frac{1}{3}\theta_{r_0} \end{aligned}$$

We estimate the probability of the corpus given this new parameters to be:

$$-\lg \frac{P\left(D|\theta'\right)}{P\left(D|\theta\right)} \approx -\lg \frac{\theta'_{r_1}\theta'_{r_2}\theta'_{r_3}}{\theta_{r_0}}$$

To generalize this to a set of rule segmentations, we construct the new parameters θ' to reflect all the changes in the set in a first pass, and then sum the differences in DL for all the rule segmentations with the new parameters in a second pass.

5 Initial ITG

The initial ITG that we start with is learned following the best bootstrapping approach reported in

Algorithm 1 Iterative rule segmenting learning driven by minimum description length.

```
 1: Φ                                   ▷ The ITG being induced
 2: repeat
 3:    δ_sum ← 0
 4:    bs ← collect_biaffixes(Φ)
 5:    bδ ← []
 6:    for all b ∈ bs do
 7:       δ ← eval_dl(b, Φ)
 8:       if δ < 0 then
 9:          bδ ← [bδ, ⟨b, δ⟩]
10:       end if
11:    end for
12:    sort_by_delta(bδ)
13:    for all ⟨b, δ⟩ ∈ bδ do
14:       δ′ ← eval_dl(b, Φ)
15:       if δ′ < 0 then
16:          Φ ← make_segmentations(b, Φ)
17:          δ_sum ← δ_sum + δ′
18:       end if
19:    end for
20: until δ_sum ≥ 0
21: return Φ
```

Algorithm 2 Iterative rule segmenting learning driven by minimum conditional description length.

```
 1: Φ, Ψ                                ▷ The auxiliary and initial ITG
 2: repeat
 3:    δ_sum ← 0
 4:    bs ← collect_biaffixes(Φ)
 5:    bδ ← []
 6:    for all b ∈ bs do
 7:       δ ← eval_cdl(b, Ψ, Φ)
 8:       if δ < 0 then
 9:          bδ ← [bδ, ⟨b, δ⟩]
10:       end if
11:    end for
12:    sort_by_delta(bδ)
13:    for all ⟨b, δ⟩ ∈ bδ do
14:       δ′ ← eval_cdl(b, Ψ, Φ)
15:       if δ′ < 0 then
16:          Φ ← make_segmentations(b, Φ)
17:          δ_sum ← δ_sum + δ′
18:       end if
19:    end for
20: until δ_sum ≥ 0
21: return Φ
```

Saers *et al.* (2012). That is: we start by initializing a token-based bracketing finite-state transduction grammar, or FSTG, parameterized with relative frequencies from the training corpus. We then tune the parameters to the training corpus, and then change the structure of the grammar to include lexical rules that can be formed by chunking adjacent preterminals. The tune–chunk step is repeated twice, before transforming the FSTG into a bracketing linear inversion transduction grammar, or LITG (Saers *et al.*, 2010), whose parameters are also tuned to the training corpus. The LITG is then transformed into a full ITG whose parameters are again tuned to the training corpus. All parameter tuning is carried out with our in-house biparser, which is based on beam search (Saers *et al.*, 2009), and expectation maximization (Dempster *et al.*, 1977). We also prune away very improbable rules to reduce noise, which makes the model perform better than reported in the original paper, providing a more solid baseline for comparison.

6 Augmenting the initial ITG

To augment the initial ITG we will search top-down for rules that the chunking approach were unable to find. We do this by initializing an auxiliary ITG that merely contains all sentence pairs as biterminals. This auxiliary ITG is then iteratively segmented until we arrive at a set of rules which cannot be segmented to further reduce the conditional description length of the auxiliary ITG given the initial ITG. The initial and auxiliary ITGs are then combined to form the augmented ITG.

Learning the auxiliary ITG is very similar to learning the baseline ITG. The motivation and initialization are identical, but rather than driving the segmentation by evaluating description length, it is driven by evaluating conditional description length (CDL). Algorithm 2 is thus very similar to Algorithm 1, except that there is an initial ITG, and that Algorithm 2 calls `eval_cdl` on lines 7 and 14, where Algorithm 1 calls `eval_dl`. To evaluate the CDL of a proposed set of candidate segmentations, we now need to calculate the difference in CDL between the current model, and the model that would result from committing to the candidate segmentations:

$$\begin{aligned} &DL\left(\Phi', D|\Psi\right) - DL\left(\Phi, D|\Psi\right) \\ &\quad = DL\left(D|\Phi', \Psi\right) - DL\left(D|\Phi, \Psi\right) \\ &\qquad + DL\left(\Phi'|\Psi\right) - DL\left(\Phi|\Psi\right) \end{aligned}$$

The model lengths are still trivial, as we merely have to encode the rules that are removed and inserted according to our encoding scheme, but we

Table 1: The results of decoding.

ITG model	BLEU	NIST	Rules
Baseline	17.44	4.3909	47,298
Initial only	15.71	4.1267	251,947
Auxiliary only	16.11	3.9334	60,133
Augmented	19.32	4.4243	301,293

still need to calculate the change in the length of the data, which is:

$$DL\left(D|\Phi',\Psi\right)-DL\left(D|\Phi,\Psi\right)=-\lg\frac{P\left(D|\Phi',\Psi\right)}{P\left(D|\Phi,\Psi\right)}$$

For the sake of convenience in efficiently calculating this probability, we make the simplifying assumption that:

$$P\left(D|\Phi,\Psi\right)\approx P\left(D|\Phi\right)=P\left(D|\theta\right)$$

where θ is the model parameters, which allow us to approximate the difference in data CDL as:

$$-\lg\frac{P\left(D|\theta'\right)}{P\left(D|\theta\right)}$$

This is the same problem that we had for the baseline model, and we solve it in the same way: by assuming probability mass to be distributed uniformly over over the new rules and by approximating the change in corpus probability in closed form.

Although this simplifying assumption is reasonable for calculating the difference in probability of the data given the augmented model, it might not be such a good assumption during decoding. So, when using the augmented model for translation, we interpolate the initial and auxiliary ITG to produce the augmented ITG. The parameters of the augmented ITG are set such that:

$$\theta_r^{\Phi,\Psi}=\alpha\theta_r^{\Phi}+(1-\alpha)\theta_r^{\Psi}$$

for all rules r, where θ is the probability of a rule under a specific ITG, and α is a weighting parameter that determine which ITG we trust more. For the experiments in this paper, we fixed $\alpha=\frac{1}{2}$.

7 Experimental setup

To test the new learning algorithm, we will induce two ITGs: one using the baseline learning algorithm and one using the presented augmenting algorithm that relies on minimizing the introduced conditional description length. We use the Chinese–English translation task from IWSLT07 (Fordyce, 2007) as training and test data. In contains 46,867 sentence pairs of training data, and 489 sentence pairs of test data with 6 reference translations each. To decode with the learned model, we use our in-house ITG decoder with a trigram language model learned on the English part of the training data. The decoder uses CKY-style parsing (Cocke, 1969; Kasami, 1965; Younger, 1967) with cube pruning to integrate the language model (Chiang, 2007). The language model is trained with SRILM (Stolcke, 2002). To evaluate the output we use BLEU (Papineni *et al.*, 2002) and NIST (Doddington, 2002).

8 Results

The results (Table 1) show the baseline ITG and the proposed augmented ITG, as well as test scores for the two intermediate steps: the initial and auxiliary ITGs. The augmented ITG is significantly better (19.32 compared to 17.44 BLEU) than the baseline ITG, but also significantly larger (301,293 compared to 47,298). The number of rules is known to be somewhat correlated with the translation quality, so it is hard to draw any conclusions from these data. The fact that the augmented ITG is significantly better than the initial ITG (19.32 compared to 15.71 BLEU) with only a modest increase in the number of rules (49,346 extra rules) is, however, very interesting. It shows that the auxiliary ITG is indeed learning rules that complement the initial ITG well. This picture is further corroborated by the fact that the auxiliary ITG is far behind the full augmented ITG in terms of translation quality.

9 Conclusion

We have presented conditional minimum description length, a theoretically well-founded learning objective particularly suited for searching for a supplemental model tailored to augmenting a preexisting model, which we have applied to the task of inducing ITGs by augmenting a bottom-up chunked inversion transduction grammar with rules obtained by iteratively splitting existing rules into smaller rules. We have further shown empirically that the proposed augmentation strategy significantly boosts the quality of an initial ITG. The model provides an obvious foundation for generalization to more abstract transduction grammars with informative nonterminals.

Acknowledgements

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. 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, the EU, or RGC.

References

Phil Blunsom and Trevor Cohn. Inducing synchronous grammars with slice sampling. In *NAACL HLT 2010*, pages 238–241, Los Angeles, California, Jun 2010.

Phil Blunsom, Trevor Cohn, and Miles Osborne. Bayesian synchronous grammar induction. In *NIPS 21*, Vancouver, Canada, Dec 2008.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. A gibbs sampler for phrasal synchronous grammar induction. In *ACL-IJCNLP 2009*, pages 782–790, Suntec, Singapore, Aug 2009.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Machine Translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

David Burkett, John Blitzer, and Dan Klein. Joint parsing and alignment with weakly synchronized grammars. In *NAACL HLT 2010*, pages 127–135, Los Angeles, California, Jun 2010.

Colin Cherry and Dekang Lin. Inversion transduction grammar for joint phrasal translation modeling. In *SSST*, pages 17–24, Rochester, New York, Apr 2007.

David Chiang, Kevin Knight, and Wei Wang. 11,001 new features for statistical machine translation. In *NAACL HLT 2009*, pages 218–226, Boulder, Colorado, Jun 2009.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *ACL-05*, pages 263–270, Ann Arbor, Michigan, Jun 2005.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007.

John Cocke. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

Arthur Pentland Dempster, Nan M. Laird, and Donald Bruce Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT '02*, pages 138–145, San Diego, California, 2002.

C. S. Fordyce. Overview of the IWSLT 2007 evaluation campaign. In *IWSLT 2007*, pages 1–12, 2007.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *COLING/ACL 2006*, pages 961–968, Sydney, Australia, Jul 2006.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. Better word alignments with supervised itg models. In *ACL-IJCNLP 2009*, pages 923–931, Suntec, Singapore, Aug 2009.

Tadao Kasami. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-00143, Air Force Cambridge Research Laboratory, 1965.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *HLT-NAACL 2003*, volume 1, pages 48–54, Edmonton, Canada, May/Jun 2003.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. An unsupervised model for joint phrase alignment and extraction. In *ACL HLT 2011*, pages 632–641, Portland, Oregon, Jun 2011.

Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara. Machine translation without words through substring alignment. In *ACL 2012*, pages 165–174, Jeju Island, Korea, Jul 2012.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic

evaluation of machine translation. In *ACL-02*, pages 311–318, Philadelphia, Pennsylvania, Jul 2002.

Jason Riesa and Daniel Marcu. Hierarchical search for word alignment. In *ACL 2010*, pages 157–166, Uppsala, Sweden, Jul 2010.

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, Jun 1983.

Markus Saers and Dekai Wu. Improving phrase-based translation via word alignments from Stochastic Inversion Transduction Grammars. In *SSST-3*, pages 28–36, Boulder, Colorado, Jun 2009.

Markus Saers and Dekai Wu. Principled induction of phrasal bilexica. In *EAMT-2011*, pages 313–320, Leuven, Belgium, May 2011.

Markus Saers, Joakim Nivre, and Dekai Wu. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *IWPT'09*, pages 29–32, Paris, France, Oct 2009.

Markus Saers, Joakim Nivre, and Dekai Wu. Word alignment with stochastic bracketing linear inversion transduction grammar. In *NAACL HLT 2010*, pages 341–344, Los Angeles, California, Jun 2010.

Markus Saers, Karteek Addanki, and Dekai Wu. From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction. In *COLING 2012*, pages 2325–2340, Mumbai, India, Dec 2012.

Markus Saers, Karteek Addanki, and Dekai Wu. Iterative rule segmentation under minimum description length for unsupervised transduction grammar induction. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing, First International Conference, SLSP 2013*, Lecture Notes in Artificial Intelligence (LNAI). Springer, Tarragona, Spain, Jul 2013.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, Jul, Oct 1948.

Ray J. Solomonoff. A new method for discovering the grammars of phrase structure languages. In *IFIP*, pages 285–289, 1959.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *ICSLP2002 - INTERSPEECH 2002*, pages 901–904, Denver, Colorado, Sep 2002.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based Word Alignment in Statistical Translation. In *COLING-96*, volume 2, pages 836–841, 1996.

Dekai Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403, 1997.

Daniel H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *ACL-08: HLT*, pages 97–105, Columbus, Ohio, Jun 2008.