What is Hidden among Translation Rules

Libin Shen
Persado
50 West 17th Street
New York, NY 10011
libin.shen@persado.com

Bowen Zhou
IBM T. J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
zhou@us.ibm.com

Abstract

Most of the machine translation systems rely on a large set of translation rules. These rules are treated as discrete and independent events. In this short paper, we propose a novel method to model rules as observed generation output of a compact hidden model, which leads to better generalization capability. We present a preliminary generative model to test this idea. Experimental results show about one point improvement on TER-BLEU over a strong baseline in Chinese-to-English translation.

1 Introduction

Most of the modern Statistical Machine Translation (SMT) systems, for example (Koehn et al., 2003; Och and Ney, 2004; Chiang, 2005; Marcu et al., 2006; Shen et al., 2008), employ a large rule set that may contain tens of millions of translation rules or even more. In these systems, each translation rule has about 20 dense features, which represent key statistics collected from the training data, such as word translation probability, phrase translation probability etc. Except for these common features, there is no connection among the translation rules. The translation rules are treated as independent events.

The use of sparse features as in (Arun and Koehn, 2007; Watanabe et al., 2007; Chiang et al., 2009) to some extent mitigated this problem. In their work, there are as many as 10,000 features defined on the appearance of certain frequent words and Part of Speech (POS) tags in rules. They provide significant improvement in automatic evaluation metrics. However, these sparse features fire quite randomly and infrequently on each rule. Thus, there is still plenty of space to better model translation rules.

In this paper, we will explore the relationship among translation rules. We no longer view rules as discrete or unrelated events. Instead, we view rules, which are observed from training data, as random variables generated by a hidden model. This generative process itself is also hidden. All possible generative processes can be represented with factorized structures such as weighted hypergraphs and finite state machines. This approach leads to a compact model that has better generalization capability and allows translation rules not explicitly observed in training date.

This paper reports work-in-progress to exploit hidden relations among rules. Preliminary experiments show about one point improvement on TER-BLEU over a strong baseline in Chinese-to-English translation.

2 Hidden Models

Let $\mathcal{G} = \{(r, f)\}$ be a grammar observed from parallel training data, where f is the frequency of a bilingual translation rule r.

Let $\mathcal{M}$ be a hidden model that generates every translation rule r. For example, $\mathcal{M}$ could be modeled with a weighted hypergraph or finite state machine. For the sake of convenience, in this section we assume $\mathcal{M}$ is a meta-grammar $\mathcal{M} = \{m\}$, where each m represents a meta-rule. For each translation r, there exists a hypergraph H_r that represents all possible derivations $D_r = \{d\}$ that can generate rule r. Here, each derivation d is a hyperpath using meta-rules M_d, where $M_d \subseteq \mathcal{M}$. Thus, we can use hypergraph H_r to characterize r. Translation rules in $\mathcal{G}$

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 839–844,
Seattle, Washington, USA, 18-21 October 2013.

can share nodes and meta-rules in their hypergraphs, so that $\mathcal{M}$ is more compact model than $\mathcal{G}$.

In the rest of this section, we will introduce three methods to quantify H_r as features of rule r. It should be noted that there are more ways to exploit the compact model of $\mathcal{M}$ than these three.

2.1 Type 1 : A Generative Model

Let θ be the parameters of a statistical model $Pr(m;\theta)$ for meta-rules m in meta-grammar $\mathcal{M}$ estimated from the observed translation grammar $\mathcal{G}$. The probability of a translation rule r can be calculated as follows.

$$\begin{aligned} Pr(r;\theta) &\propto Pr(H_r;\theta) \\ &= \sum_{d \in D_r} Pr(d;\theta) \end{aligned} \quad (1)$$

By assuming separability,

$$Pr(d;\theta) = \prod_{m \in M_d} Pr(m;\theta) \quad (2)$$

we can further decompose rule probability $Pr(r;\theta)$ as below.

$$Pr(r;\theta) = \sum_{d \in D_r} \prod_{m \in M_d} Pr(m;\theta) \quad (3)$$

In practice, $Pr(r;\theta)$ in (3) can be calculated through bottom-up dynamic programming on hypergraph H_r. Hypergraphs of different rules can share nodes and meta-rules. This reveals the underlying relationship among translation rules.

As a by-product of this generative model, we use the log-likelihood of a translation rule, $\log Pr(r;\theta)$, as a new dense feature. We call it *Type 1* in experiments.

2.2 Type 2 : Meta-Rules as Sparse Features

As given in (3), likelihood of a translation rule is a function over $Pr(m;\theta)$, in which θ is estimated from the training data with a generative model. Previous work in (Chiang et al., 2009) showed the advantage of using a discriminative model to optimize individual weights for these factors towards a better automatic score.

Following this practice, we treat each meta-rule m as a sparse feature. Feature value $f(m) = 1$ if and only if m is used in hypergraph H_r. Otherwise, its default value is 0. We call these features *Type 2* in experiments. The Type 2 system contains the log-likelihood feature in Type 1.

2.3 Type 3 : Posterior as Feature Values

A natural question on the binary sparse features defined above is why all the active features have the same value of 1. We use these meta-rules to represent a translation rule in feature space. Intuitively, for meta-rules with closer connection to the translation rules, we hope to use relatively larger feature values to increase their effect.

We formalize this intuition with the posterior probability that a meta-rule m is used to generate r, as below.

$$\begin{aligned} f(m) &\equiv Pr(m|r;\theta) \\ &= \frac{Pr(m,r;\theta)}{Pr(r;\theta)} \\ &= \frac{\sum_{d \in D_r, m \in M_d} Pr(d;\theta)}{Pr(r;\theta)} \end{aligned} \quad (4)$$

The posterior in (4) could be too sharp. Following the common practice, we smooth the posterior features with a scaling factor α.

$$f(m) \equiv Pr(m|r)^{\alpha}$$

We use *Type 3*(α) to represent the posterior model with a scaling factor of α in experiments. The Type 3 systems also contain the log-likelihood feature in Type 1.

2.4 Parameter Estimation

Now we explain how to obtain parameter θ. With proper definition of the underlying model $\mathcal{M}$, we can estimate θ with the traditional EM algorithm or Bayesian methods.

In the next section, we will present an example of the hidden model. We will employ the EM algorithm to estimate the parameters in θ. Here, translation rules and their frequencies in $\mathcal{G}$ are observed data, and derivation d for each rule r is hidden. At the *Expectation* step, we search all derivations d in D_r of each rule r and calculate their probabilities according to equation (2). At the *Maximization* step, we re-estimate θ on all derivations in proportion to their posterior probability.

3 Case Study

In Section 2, we explored the use of meta-grammars as the underlying model $\mathcal{M}$ and developed three methods to define features. Similar techniques can be applied to finite state machines and other underlying models. Now, we introduce a POS-based underlying model to illustrate the generic model proposed in Section 2. We will show experimental results in Section 4.

3.1 Meta-rules on POS tags

Let $r \in \mathcal{G}$ be a translation rule composed of a pair of source and target word strings (F_w, E_w). Let F_p and E_p be the POS tags for the source and target sides respectively. For the sake of simplicity as the first attempt, we treat non-terminal as a special word X with POS tag X.

Suppose we have a Chinese-to-English translation rule as below.

$$yuehan\ qu\ zhijiage \Rightarrow john\ leaves\ for\ chicago$$

We call

$$NR\ VV\ NR \Rightarrow NNP\ VBZ\ IN\ NNP \quad (5)$$

a translation rule in POS tags.

We will propose an underlying model $\mathcal{M}$ to generate translation rules in POS tags instead of translation rules themselves. For the rest of this section, we take translation rules in POS tags as the target of our generative model. We define meta-rules on pairs of POS tag strings, e.g. $NR\ VV \Rightarrow NNP\ VBZ$.

We can decompose the probability of translation rule in (5) into a product on meta-rule probabilities via various derivations, such as

- $Pr(NR\ VV,\ NNP\ VBZ) \times Pr(NR,\ IN\ NNP)$, and
- $Pr(NR,\ NNP) \times Pr(VV,\ VBZ\ IN) \times Pr(NR,\ NNP)$.

3.2 The Underlying Model and Features

Now, we introduce a generative model M for translation rules in POS tags. We still use the example in (5) as shown in Figure 1, where the top box represents the source side and the bottom box represents the target side. Dotted lines represent word alignments on three pairs of words.

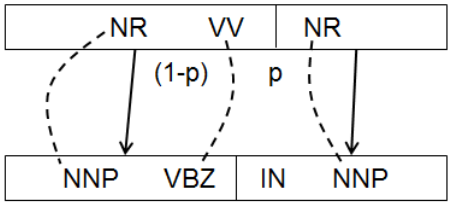

Figure 1: An example

We first generate the number of source tokens of a translation rule with a uniform distribution for up to, for example, 7 tokens.

Then we split the source side into chunks with a binomial distribution with a Bernoulli variable at the gap between each two continuous words, which splits the two words into two chunks with a probability of p. For example, the probability of obtaining two chunks *NR VV* and *NR* is $(1-p)p$, as shown in Figure 1.

Suppose we split the target side into two parts, *NNP VBZ* and *IN NNP*, which respects the word alignments. It generates two meta-rules *NR VV* $\Rightarrow$ *NNP VBZ* and *NR* $\Rightarrow$ *IN NNP*, as shown in Figure 1. The probability for the first meta-rule is

$$Pr(|E| = 2 \mid |F| = 2) \times$$

$$Pr(NR\ VV, NNP\ VBZ \mid |F| = 2, |E| = 2),$$

where $|F|$ represents the number of source tokens, and $|E|$ the number of target tokens. Similarly, the probability of the second one is as follows.

$$Pr(|E| = 2 \mid |F| = 1) \times$$

$$Pr(NR, IN\ NNP \mid |F| = 1, |E| = 2).$$

To sum up, the probability of a derivation d for a translation rule $r : F \Rightarrow E$ is

$$\begin{aligned} Pr(d) \approx\ & Pr_{\theta_1}(|F|) \\ & \times Pr_{\theta_2}(F_s) \\ & \times \prod_{m \in M_d} Pr_{\theta_3}(|E_m| \mid |F_m|) \\ & \times \prod_{m \in M_d} Pr_{\theta_4}(m \mid |F_m|, |E_m|) \end{aligned} \quad (6)$$

where F_m and E_m are source and target sides of a meta-rule m used in derivation d, and F_s is a splitting of the source side. As for the distributions, we

have

$$\begin{aligned}\theta_1 &\sim Uniform \\ \theta_2 &\sim Binomial \\ \theta_3 &\sim Categorical \\ \theta_4 &\sim Categorical\end{aligned}$$

where θ_1 and θ_2 have pre-selected hyperparameters, and θ_3 and θ_4 are estimated with the EM algorithm.

As for sparse features, we will obtain 7 meta-rule features as below.

- *NR* $\Rightarrow$ *NNP*
- *VV* $\Rightarrow$ *VBZ*
- *VV* $\Rightarrow$ *VBZ IN*
- *NR VV* $\Rightarrow$ *NNP VBZ*
- *NR VV* $\Rightarrow$ *NNP VBZ IN*
- *VV NR* $\Rightarrow$ *VBZ IN NNP*
- *NR VV NR* $\Rightarrow$ *NNP VBZ IN NNP*

All of them respect the word alignment, which means that

- there is no alignment that aligns one word in a meta-rule with the other out of the same meta-rule, and
- there is at least one alignment within a meta-rule.

3.3 Implementation Details

Even though the size of all possible meta-rules is much smaller than the space of translation rules, it is still too large to work with existing optimization methods for sparse features in MT, i.e. MIRA (Chiang et al., 2009) or L-BFGS (Matsoukas et al., 2009). In practice, we have to limit the feature space to around 20,000 dimensions.

For this purpose, we first use a frequency based method to filter meta-rule features. Specifically, we first divide all the meta-rules into 100 bins, $(|F|, |E|)$, where $|F|$ is the number of words on the source side, and $|E|$ the target side, $0 < |F|, |E| \leq 10$. For each bin, we keep the same top k-percentile of the meta-rules such that we obtain a total of 20,000 meta-rules as features.

System	BLEU%	TER%	T-B
Baseline	30.35	55.32	24.97
Type 1	30.74	55.48	24.74
Type 2	31.07	55.07	24.00
Type 3 (1)	30.93	55.34	24.41
Type 3 (0.1)	31.05	55.02	23.97
Type 3 (0.01)	31.09	54.96	23.87

Table 1: scores on test-1

A shortcoming of this filtering method is that all these features are positive indicators, while low-frequency negative indicators are discarded. In order to keep the features of various level of frequency, we define class features with a 3-tuple $C(|F|, |E|, q)$, where $|F|$ and $|E|$ are numbers of source and target words as defined above, and q is the integer part of the log_2 value of the feature frequency in the training data.

In this way, each meta-rule feature can be mapped to one of these classes. The value of a class feature equals the sum of the meta-rule features that mapped into this class. We have about 2,000 class features defined in this way. They are applied on both Type 2 and Type 3 features.

4 Experiments

We carry out our experiments on web genre of Chinese-to-English translation. The training set contains about 10 million parallel sentences available to Phase 1 of the DARPA BOLT MT task. The tune set contains 1275 sentences. Each has four references. There are two test sets. Test-1 is from a similar source of the tune set, and it contains 1239 sentences. Test-2 is the web part of the MT08 evaluation data.

Our baseline system is a home-made Hiero (Chiang, 2005) style system. The baseline rule set contains about 17 million rules. It contains about 40 dense features, including a 6-gram LM.

The sparse feature optimization algorithm is similar to the MIRA recipe described in (Chiang et al., 2009). We optimize on TER-BLEU (Snover et al., 2006; Papineni et al., 2001).

The BLEU, TER and T-B scores on the two tests are shown in Tables 1 and 2. It should be noted that, even though our metric of tuning is T-B, the baseline

System	BLEU%	TER%	T-B
Baseline	25.80	56.96	31.16
Type 1	26.18	57.09	30.91
Type 2	26.63	56.64	30.01
Type 3 (1)	26.30	57.00	30.70
Type 3 (0.1)	26.34	56.73	30.39
Type 3 (0.01)	26.50	56.73	30.23

Table 2: scores on test-2 (MT08-WB)

system already provides a very competitive BLEU score on MT08-WB as compared the best system in the evaluation[1], thanks to comprehensive features in the baseline system and more data in training.

All the three types of systems provide consistent improvement on both test sets in terms of T-B, our optimization metric. Type 1 gives marginal improvement of 0.2. This shows the limitation of the generative feature. When we use meta-rules as binary sparse features in Type 2, we obtain about one point improvement on T-B on both sets. This shows the advantage of tuning individual meta-rule weights over a generative model. Type 3 (0.01) and Type 2 are at the same level. Proper smoothing is important to Type 3.

5 Discussion

In the case study of Section 3, we use POS-based rules as hidden states. However, it should be noted that the hidden structures surely do not have to be POS tags. For example, an alternative could be unsupervised NT splitting similar to (Huang et al., 2010).

The meta-grammar based approach was also motivated by the insight acquired on mono-lingual linguistic grammar generation, especially in the TAG related research (Xia, 2001; Prolo, 2002). Meta-grammar was viewed as an effective way to remove redundancy in grammars.

The link between Tree Adjoining Grammar (TAG) (Joshi et al., 1975; Joshi and Schabes, 1997) and MT was first introduced in (Shieber and Schabes, 1990), a pioneer work in tree-to-tree translation. (DeNeefe and Knight, 2009) re-visited the use of adjoining operation in the context of Statistical MT, and reported encouraging results. On the other hand, (Dras, 1999) showed how a meta-level grammar could help in modeling parallel operations in (Shieber and Schabes, 1990). Our work is another effort of statistical modeling of well-recognized linguistic insight in NLP and MT.

[1] http://www.itl.nist.gov/iad/mig/tests/mt/2008/

6 Conclusions and Future Work

In this paper, we introduced a novel method to model translation rules as observed generation output of a compact hidden model. As a case study to capitalize this model, we presented three methods to enrich rule modeling with features defined on a hidden model. Preliminary experiments verified gain of one point on TER-BLEU over a strong baseline in Chinese-to-English translation.

As for future work, we plan to extend this work in the following aspects.

- To try other prior distributions to generate the number of source tokens.
- Unsupervised and semi-supervised learning of hidden models.
- To incorporate rich models into the generative process, e.g. reordering, non-terminals, structural information and lexical models.
- To improve the posterior model with better parameter estimation, e.g. Bayesian methods.
- To replace the exhaustive translation rule set with a compact meta grammar that can create and parameterize new translation rules dynamically, which is the ultimate goal of this line of work.

Acknowledgments

We would like thank the anonymous reviewers for their valuable comments. Haitao Mi and Martin Cmejrek kindly helped on data preparation.

This work was done when the first author was at IBM. The work was supported by DARPA under Grant HR0011-12-C-0015 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the ofcial views or policies, either expressed or implied, of the DARPA.

References

Abhishek Arun and Philipp Koehn. 2007. Online learning methods for discriminative training of phrase based statistical machine translation. In *Proceedings of MT Summit XI*.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of the 2009 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 263–270, Ann Arbor, MI.

Steve DeNeefe and Kevin Knight. 2009. Synchronous tree adjoining machine translation. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing*, pages 727–736, Singapore.

Mark Dras. 1999. A meta-level grammar: redefining synchronous tag for translation and paraphrase. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhongqiang Huang, Martin Cmejrek, and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proceedings of the 2010 Conference of Empirical Methods in Natural Language Processing*.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer-Verlag.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference of Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia.

Spyros Matsoukas, Antti-Veikko Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing*.

Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

Kishore Papineni, Salim Roukos, and Todd Ward. 2001. Bleu: a method for automatic evaluation of machine translation. IBM Research Report, RC22176.

Carlos Prolo. 2002. Generating the xtag english grammar using metarules. In *Proceedings of the 19th international conference on Computational linguistics (COLING)*.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Stuart Shieber and Yves Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of COLING '90: The 13th Int. Conf. on Computational Linguistics*, pages 253–258, Helsinki, Finland.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

T. Watanabe, J. Suzuki, H. Tsukuda, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Conference of Empirical Methods in Natural Language Processing*.

F. Xia. 2001. *Automatic Grammar Generation From Two Different Perspectives*. Ph.D. thesis, University of Pennsylvania.