Statistical Machine Translation with a Factorized Grammar

Libin Shen and **Bing Zhang** and **Spyros Matsoukas** and
Jinxi Xu and **Ralph Weischedel**
Raytheon BBN Technologies
Cambridge, MA 02138, USA
{lshen,bzhang,smatsouk,jxu,weisched}@bbn.com

Abstract

In modern machine translation practice, a statistical phrasal or hierarchical translation system usually relies on a huge set of translation rules extracted from bi-lingual training data. This approach not only results in space and efficiency issues, but also suffers from the sparse data problem. In this paper, we propose to use factorized grammars, an idea widely accepted in the field of linguistic grammar construction, to generalize translation rules, so as to solve these two problems. We designed a method to take advantage of the XTAG English Grammar to facilitate the extraction of factorized rules. We experimented on various setups of low-resource language translation, and showed consistent significant improvement in BLEU over state-of-the-art string-to-dependency baseline systems with 200K words of bi-lingual training data.

1 Introduction

A statistical phrasal (Koehn et al., 2003; Och and Ney, 2004) or hierarchical (Chiang, 2005; Marcu et al., 2006) machine translation system usually relies on a very large set of translation rules extracted from bi-lingual training data with heuristic methods on word alignment results. According to our own experience, we obtain about 200GB of rules from training data of about 50M words on each side. This immediately becomes an engineering challenge on space and search efficiency.

A common practice to circumvent this problem is to filter the rules based on development sets in the step of rule extraction or before the decoding phrase, instead of building a real distributed system. However, this strategy only works for research systems, for which the segments for translation are always fixed.

However, do we really need such a large rule set to represent information from the training data of much smaller size? Linguists in the grammar construction field already showed us a perfect solution to a similar problem. The answer is to use a factorized grammar. Linguists decompose lexicalized linguistic structures into two parts, (unlexicalized) templates and lexical items. Templates are further organized into families. Each family is associated with a set of lexical items which can be used to lexicalize all the templates in this family. For example, the XTAG English Grammar (XTAG-Group, 2001), a hand-crafted grammar based on the Tree Adjoining Grammar (TAG) (Joshi and Schabes, 1997) formalism, is a grammar of this kind, which employs factorization with LTAG e-tree templates and lexical items.

Factorized grammars not only relieve the burden on space and search, but also alleviate the sparse data problem, especially for low-resource language translation with few training data. With a factored model, we do not need to observe exact "template – lexical item" occurrences in training. New rules can be generated from template families and lexical items either offline or on the fly, explicitly or implicitly. In fact, the factorization approach has been successfully applied on the morphological level in previous study on MT (Koehn and Hoang, 2007). In this work, we will go further to investigate factorization of rule structures by exploiting the rich XTAG English Grammar.

We evaluate the effect of using factorized translation grammars on various setups of low-resource language translation, since low-resource MT suffers greatly on poor generalization capability of trans-

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 616–625,
MIT, Massachusetts, USA, 9-11 October 2010.

lation rules. With the help of high-level linguistic knowledge for generalization, factorized grammars provide consistent significant improvement in BLEU (Papineni et al., 2001) over string-to-dependency baseline systems with 200K words of bi-lingual training data.

This work also closes the gap between compact hand-crafted translation rules and large-scale unorganized automatic rules. This may lead to a more effective and efficient statistical translation model that could better leverage generic linguistic knowledge in MT.

In the rest of this paper, we will first provide a short description of our baseline system in Section 2. Then, we will introduce factorized translation grammars in Section 3. We will illustrate the use of the XTAG English Grammar to facilitate the extraction of factorized rules in Section 4. Implementation details are provided in Section 5. Experimental results are reported in Section 6.

2 A Baseline String-to-Tree Model

As the baseline of our new algorithm, we use a string-to-dependency system as described in (Shen et al., 2008). There are several reasons why we take this model as our baseline. First, it uses syntactic tree structures on the target side, which makes it easy to exploit linguistic information. Second, dependency structures are relatively easier to implement, as compared to phrase structure grammars. Third, a string-to-dependency system provides state-of-the-art performance on translation accuracy, so that improvement over such a system will be more convincing.

Here, we provide a brief description of the baseline string-to-dependency system, for the sake of completeness. Readers can refer to (Shen et al., 2008; Shen et al., 2009) for related information.

In the baseline string-to-dependency model, each translation rule is composed of two parts, source and target. The source sides is a string rewriting rule, and the target side is a tree rewriting rule. Both sides can contain non-terminals, and source and target non-terminals are one-to-one aligned. Thus, in the decoding phase, non-terminal replacement for both sides are synchronized.

Decoding is solved with a generic chart parsing algorithm. The source side of a translation rule is used to detect when this rule can be applied. The target side of the rule provides a hypothesis tree structure for the matched span. Mono-lingual parsing can be viewed as a special case of this generic algorithm, for which the source string is a projection of the target tree structure.

Figure 1 shows three examples of string-to-dependency translation rules. For the sake of convenience, we use English for both source and target. Upper-cased words represent source, while lower-cased words represent target. X is used for non-terminals for both sides, and non-terminal alignment is represented with subscripts.

In Figure 1, the top boxes mean the source side, and the bottom boxes mean the target side. As for the third rule, *FUN_Q* stands for a function word in the source language that represents a question.

3 Translation with a Factorized Grammar

We continue with the example rules in Figure 1. Suppose, we have "... *HATE* ... *FUN_Q*" in a given test segment. There is no rule having both *HATE* and *FUN_Q* on its source side. Therefore, we have to translate these two source words separately. For example, we may use the second rule in Figure 1. Thus, *HATE* will be translated into *hates*, which is wrong.

Intuitively, we would like to have translation rule that tell us how to translate *X1 HATE X2 FUN_Q* as in Figure 2. It is not available directly from the training data. However, if we obtain the three rules in Figure 1, we are able to predict this missing rule. Furthermore, if we know *like* and *hate* are in the same syntactic/semantic class in the source or target language, we will be very confident on the validity of this hypothesis rule.

Now, we propose a factorized grammar to solve this generalization problem. In addition, translation rules represented with the new formalism will be more compact.

3.1 Factorized Rules

We decompose a translation rule into two parts, a pair of **lexical items** and an unlexicalized **template**. It is similar to the solution in the XTAG English Grammar (XTAG-Group, 2001), while here we

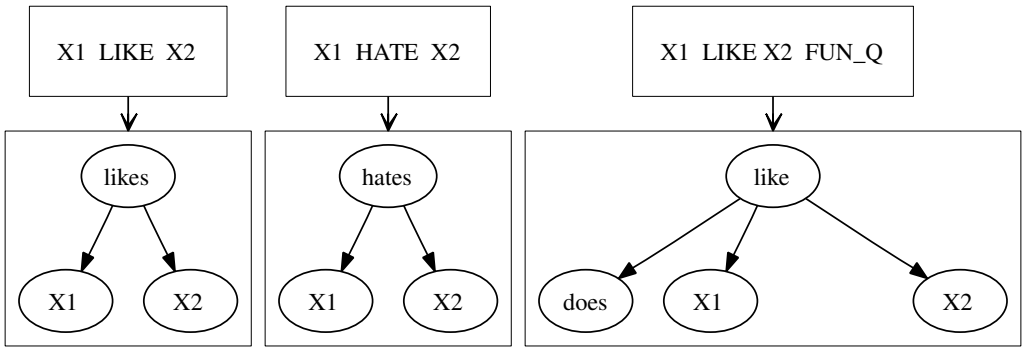

Figure 1: Three examples of string-to-dependency translation rules.

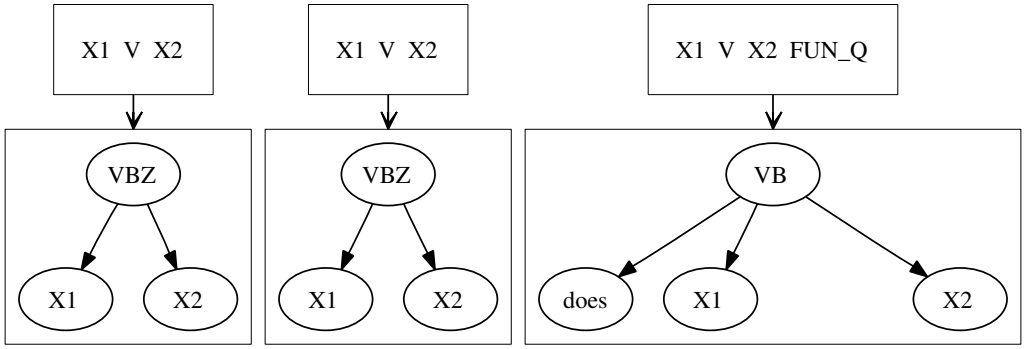

Figure 3: Templates for rules in Figure 1.

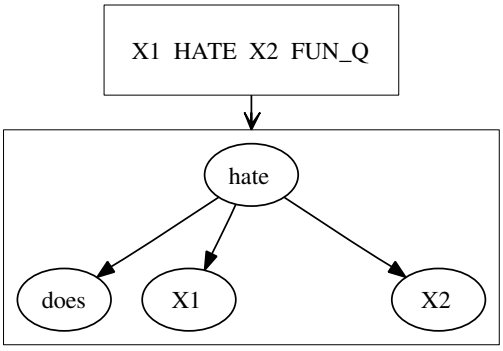

Figure 2: An example of a missing rule.

work on two languages at the same time.

For each rule, we first detect a pair of aligned head words. Then, we extract the stems of this word pair as lexical items, and replace them with their POS tags in the rule. Thus, the original rule becomes an unlexicalized rule template.

As for the three example rules in Figure 1, we will extract lexical items (*LIKE, like*), (*HATE, hate*) and (*LIKE, like*) respectively. We obtain the same lexical items from the first and the third rules.

The resultant templates are shown in Figure 3. Here, *V* represents a verb on the source side, VB stands for a verb in the base form, and *VBZ* means a verb in the third person singular present form as in the Penn Treebank representation (Marcus et al., 1994).

In the XTAG English Grammar, tree templates for transitive verbs are grouped into a family. All transitive verbs are associated with this family. Here, we assume that the rule templates representing structural variations of the same word class can also be organized into a template **family**. For example, as shown in Figure 4, templates and lexical items are associated with families. It should be noted that a template or a lexical item can be associated with more than one family.

Another level of indirection like this provides more generalization capability. As for the missing

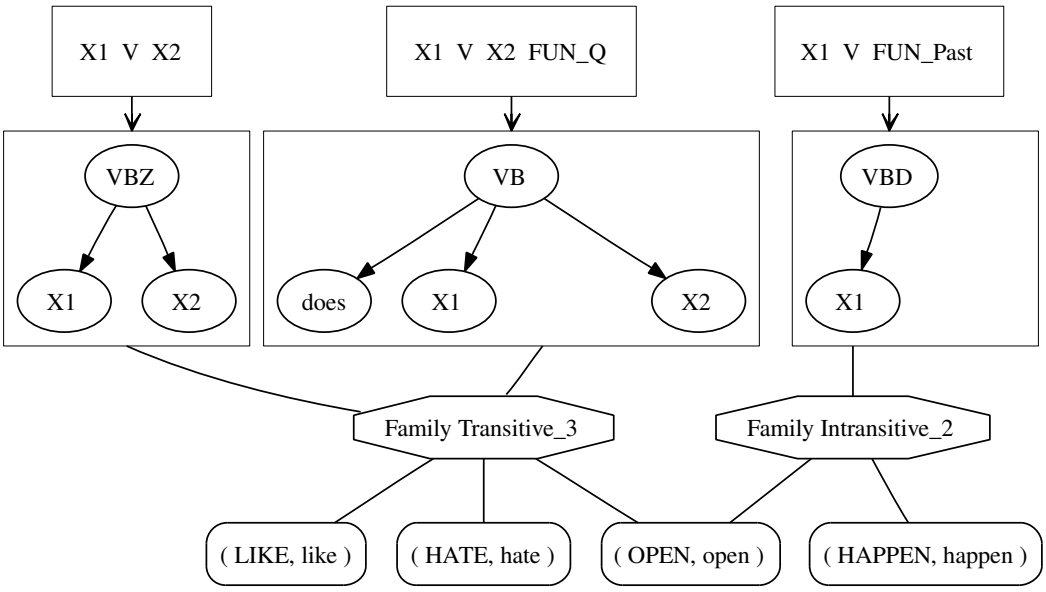

Figure 4: Templates and lexical items are associated with families.

rule in Figure 2, we can now generate it by replacing the POS tags in the second template of Figure 4 with lexical items (*HATE, hate*) with their correct inflections. Both the template and the lexical items here are associated with the family Transitive_3..

3.2 Statistical Models

Another level of indirection also leads to a desirable back-off model. We decompose a rule R into to two parts, its template P_R and its lexical items L_R. Assuming they are independent, then we can compute $Pr(R)$ as

$$Pr(R) = Pr(P_R)Pr(L_R), \text{ or}$$
$$Pr(R) = \sum_F Pr(P_R|F)Pr(L_R|F)Pr(F), \quad (1)$$

if they are conditionally independent for each family F. In this way, we can have a good estimate for rules that do not appear in the training data. The second generative model will also be useful for unsupervised learning of families and related probabilities.

In this paper, we approximate families by using target (English) side linguistic knowledge as what we will explain in Section 4, so this changes the definition of the task. In short, we will be given a list of families. We will also be given an association table $B(L, F)$ for lexical items L and families F, such that $B(L, F) = true$ if and only L is associated with F, but we do not know the distributions.

Let S be the source side of a rule or a rule template, T the target side of a rule of a rule template. We define Pr_b, the back-off conditional model of templates, as follows.

$$Pr_b(P_S|P_T, L) = \frac{\sum_{F:B(L,F)} \#(P_S, P_T, F)}{\sum_{F:B(L,F)} \#(P_T, F)}, \quad (2)$$

where $\#$ stands for the count of events.

Let P and L be the template and lexical items of R respectively. Let Pr_t be the MLE model obtained from the training data. The smoothed probability is then defined as follows.

$$\begin{aligned} Pr(R_S|R_T) &= (1-\alpha)Pr_t(R_S|R_T) \\ &\quad +\alpha Pr_b(P_S|P_T, L), \quad (3) \end{aligned}$$

where α is a parameter. We fix it to 0.1 in later experiments. Conditional probability $Pr(R_T|R_S)$ is defined in a similar way.

3.3 Discussion

The factorized models discussed in the previous section can greatly alleviate the sparse data problem, especially for low-resource translation tasks. However, when the training data is small, it is not easy to

learn families. Therefore, to use unsupervised learning with a model like (1) somehow reduces a hard translation problem to another one of the same difficulty, when the training data is small.

However, in many cases, we do have extra information that we can take advantage of. For example, if the target language has rich resources, although the source language is a low-density one, we can exploit the linguistic knowledge on the target side, and carry it over to bi-lingual structures of the translation model. The setup of X-to-English translation tasks is just like this. This will be the topic of the next section. We leave unsupervised learning of factorized translation grammars for future research.

4 Using A Mono-Lingual Grammar

In this section, we will focus on X-to-English translation, and explain how to use English resources to build a factorized translation grammar. Although we use English as an example, this approach can be applied to any language pairs that have certain linguistic resources on one side.

As shown in Figure 4, intuitively, the families are intersection of the word families of the two languages involved, which means that they are refinement of the English word families. For example, a sub-set of the English transitive families may be translated in the same way, so they share the same set of templates. This is why we named the two families Transitive_3 and Intransitive_2 in Figure 4.

Therefore, we approximate bi-lingual families with English families first. In future, we can use them as the initial values for unsupervised learning.

In order to learn English families, we need to take away the source side information in Figure 4, and we end up with a template–family–word graph as shown in Figure 5. We can learn this model on large mono-lingual data if necessary.

What is very interesting is that there already exists a hand-crafted solution for this model. This is the XTAG English Grammar (XTAG-Group, 2001).

The XTAG English Grammar is a large-scale English grammar based on the TAG formalism extended with lexicalization and unification-based feature structures. It consists of morphological, syntactic, and tree databases. The syntactic database contains the information that we have represented in Figure 5 and many other useful linguistic annotations, e.g. features.

The XTAG English grammar contains 1,004 templates, organized in 53 families, and 221 individual templates. About 30,000 lexical items are associated with these families and individual templates [1]. In addition, it also has the richest English morphological lexicon with 317,000 inflected items derived from 90,000 stems. We use this resource to predict POS tags and inflections of lexical items.

In our applications, we select all the verb families plus one each for nouns, adjectives and adverbs. We use the families of the English word as the families of bi-lingual lexical items. Therefore, we have a list of about 20 families and an association table as described in Section 3.2. Of course, one can use other linguistic resources if similar family information is provided, e.g. VerbNet (Kipper et al., 2006) or WordNet (Fellbaum, 1998).

5 Implementation

Nowadays, machine translation systems become more and more complicated. It takes time to write a decoder from scratch and hook it with various modules, so it is not the best solution for research purpose. A common practice is to reduce a new translation model to an old one, so that we can use an existing system, and see the effect of the new model quickly. For example, the tree-based model proposed in (Carreras and Collins, 2009) used a phrasal decoder for sub-clause translation, and recently, DeNeefe and Knight (2009) reduced a TAG-based translation model to a CFG-based model by applying all possible adjunction operations offline and stored the results as rules, which were then used by an existing syntax-based decoder.

Here, we use a similar method. Instead of building a new decoder that uses factorized grammars, we reduce factorized rules to baseline string-to-dependency rules by performing combination of templates and lexical items in an offline mode. This is similar to the rule generation method in (DeNeefe and Knight, 2009). The procedure is as follows.

In the rule extraction phase, we first extract all the string-to-dependency rules with the baseline system.

[1] More information about XTAG is available online at http://www.cis.upenn.edu/~xtag .

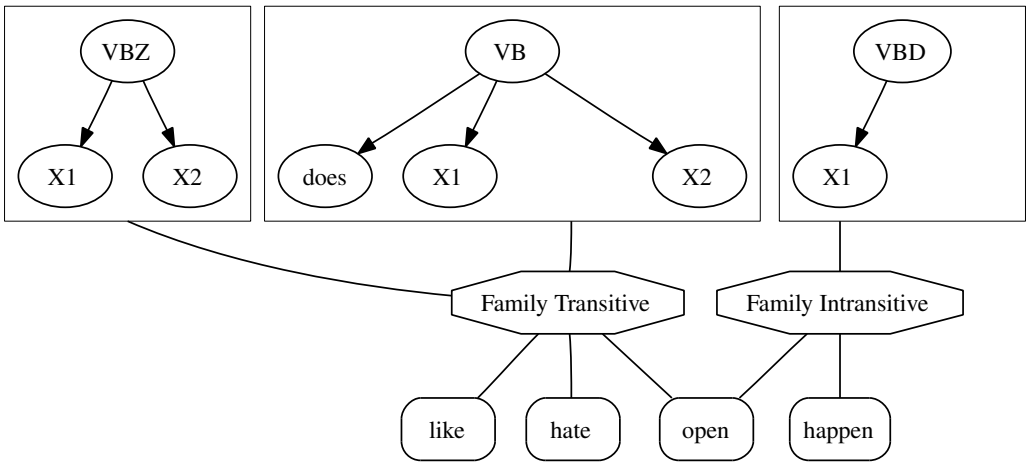

Figure 5: Templates, families, and words in the XTAG English Grammar.

For each extracted rule, we try to split it into various "template–lexical item" pairs by choosing different aligned words for delexicalization, which turns rules in Figure 1 into lexical items and templates in Figure 3. Events of templates and lexical items are counted according to the family of the target English word. If an English word is associated with more than one family, the count is distributed uniformly among these families. In this way, we collect sufficient statistics for the back-off model in (2).

For each family, we keep the top 200 most frequent templates. Then, we apply them to all the lexical items in this families, and save the generated rules. We merge the new rules with the original one. The conditional probabilities for the rules in the combined set is smoothed according to (2) and (3).

Obviously, using only the 200 most frequent templates for each family is just a rough approximation. An exact implementation of a new decoder for factorized grammars can make better use of all the templates. However, the experiments will show that even an approximation like this can already provide significant improvement on small training data sets, i.e. with no more than 2M words.

Since we implement template application in an offline mode, we can use exactly the same decoding and optimization algorithms as the baseline. The decoder is a generic chart parsing algorithm that generates target dependency trees from source string input. The optimizer is an L-BFGS algorithm that maximizes expected BLEU scores on n-best hypotheses (Devlin, 2009).

6 Experiments on Low-Resource Setups

We tested the performance of using factorized grammars on low-resource MT setups. As what we noted above, the sparse data problem is a major issue when there is not enough training data. This is one of the cases that a factorized grammar would help.

We did not tested on real low-resource languages. Instead, we mimic the low-resource setup with two of the most frequently used language pairs, Arabic-to-English and Chinese-to-English, on newswire and web genres. Experiments on these setups will be reported in Section 6.1. Working on a language which actually has more resources allows us to study the effect of training data size. This will be reported in Section 6.2. In Section 6.3, we will show examples of templates learned from the Arabic-to-English training data.

6.1 Languages and Genres

The Arabic-to-English training data contains about 200K (target) words randomly selected from an LDC corpus, LDC2006G05 A2E set, plus an Arabic-English dictionary with about 89K items. We build our development sets from GALE P4 sets. There are one tune set and two test sets for the MT systems [2]. TEST-1 has about 5000 segments and TEST-2 has about 3000 segments.

[2]One of the two test sets will later be used to tune an MT combination system.

MODEL	TUNE			TEST-1			TEST-2		
	BLEU	%BL	MET	BLEU	%BL	MET	BLEU	%BL	MET
Arabic-to-English newswire									
baseline	21.07	12.41	43.77	19.96	11.42	42.79	21.09	11.03	43.74
factorized	21.70	13.17	44.85	20.52	11.70	43.83	21.36	11.77	44.72
Arabic-to-English web									
baseline	10.26	5.02	32.78	9.40	4.87	31.26	14.11	7.34	35.93
factorized	10.67	5.34	33.83	9.74	5.20	32.52	14.66	7.69	37.11
Chinese-to-English newswire									
baseline	13.17	8.04	44.70	19.62	9.32	48.60	14.53	6.82	45.34
factorized	13.91	8.09	45.03	20.48	9.70	48.61	15.16	7.37	45.31
Chinese-to-English web									
baseline	11.52	5.96	42.18	11.44	6.07	41.90	9.83	4.66	39.71
factorized	11.98	6.31	42.84	11.72	5.88	42.55	10.25	5.34	40.34

Table 1: Experimental results on Arabic-to-English / Chinese-to-English newswire and web data. **%BL** stands for BLEU scores for documents whose BLEU scores are in the bottom 75% to 90% range of all documents. **MET** stands for METEOR scores.

The Chinese-to-English training data contains about 200K (target) words randomly selected from LDC2006G05 C2E set, plus a Chinese-English dictionary (LDC2002L27) with about 68K items. The development data setup is similar to that of Arabic-to-English experiments.

Chinese-to-English translation is from a morphology poor language to a morphology rich language, while Arabic-to-English translation is in the opposite direction. It will be interesting to see if factorized grammars help on both cases. Furthermore, we also test on two genres, newswire and web, for both languages.

Table 1 lists the experimental results of all the four conditions. The tuning metric is expected BLEU. We are also interested in the BLEU scores for documents whose BLEU scores are in the bottom 75% to 90% range of all documents. We mark it as %BL in the table. This metric represents how a system performances on difficult documents. It is important to certain percentile evaluations. We also measure METEOR (Banerjee and Lavie, 2005) scores for all systems.

The system using factorized grammars shows BLEU improvement in all conditions. We measure the significance of BLEU improvement with paired bootstrap resampling as described by (Koehn, 2004). All the BLEU improvements are over 95% confidence level. The new system also improves %BL and METEOR in most of the cases.

6.2 Training Data Size

The experiments to be presented in this section are designed to measure the effect of training data size. We select Arabic web for this set of experiments. Since the original Arabic-to-English training data LDC2006G05 is a small one, we switch to LDC2006E25, which has about 3.5M target words in total. We randomly select 125K, 250K, 500K, 1M and 2M sub-sets from the whole data set. A larger one always includes a smaller one. We still tune on expected BLEU, and test on BLEU, %BL and METEOR.

The average BLEU improvement on test sets is about 0.6 on the 125K set, but it gradually diminishes. For better observation, we draw the curves of BLEU improvement along with significance test results for each training set. As shown in Figure 6 and 7, more improvement is observed with fewer training data. This fits well with fact that the baseline MT model suffers more on the sparse data problem with smaller training data. The reason why the improvement diminishes on the full data set could be that the rough approximation with 200 most frequent templates cannot fully take advantage of this paradigm, which will be discussed in the next section.

MODEL	SIZE	TUNE			TEST-1			TEST-2		
		BLEU	%BL	MET	BLEU	%BL	MET	BLEU	%BL	MET
Arabic-to-English web										
baseline	125K	8.54	2.96	28.87	7.41	2.82	26.95	11.29	5.06	31.37
factorized		8.99	3.44	30.40	7.92	3.57	28.63	12.04	6.06	32.87
baseline	250K	10.18	4.70	32.21	8.94	4.35	30.31	13.71	6.93	35.14
factorized		10.57	4.96	33.22	9.34	4.78	31.51	14.02	7.28	36.25
baseline	500K	12.18	5.84	35.59	10.82	5.77	33.62	16.48	8.30	38.73
factorized		12.40	6.01	36.15	11.14	5.96	34.38	16.76	8.53	39.27
baseline	1M	13.95	7.17	38.49	12.48	7.12	36.56	18.86	10.00	42.18
factorized		14.14	7.41	38.99	12.66	7.34	37.14	19.11	10.29	42.56
baseline	2M	15.74	8.38	41.15	14.18	8.17	39.26	20.96	11.95	45.18
factorized		15.92	8.81	41.51	14.34	8.25	39.68	21.42	12.05	45.51
baseline	3.5M	16.95	9.76	43.03	15.47	9.08	41.28	22.83	13.24	47.05
factorized		17.07	9.99	43.18	15.49	8.77	41.41	22.72	13.10	47.23

Table 2: Experimental results on Arabic web. **%BL** stands for BLEU scores for documents whose BLEU scores are in the bottom 75% to 90% range of all documents. **MET** stands for METEOR scores.

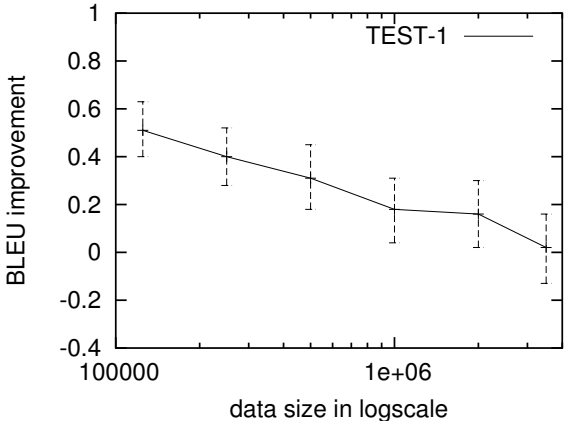

Figure 6: BLEU Improvement with 95% confidence range by using factorized grammars on TEST-1.

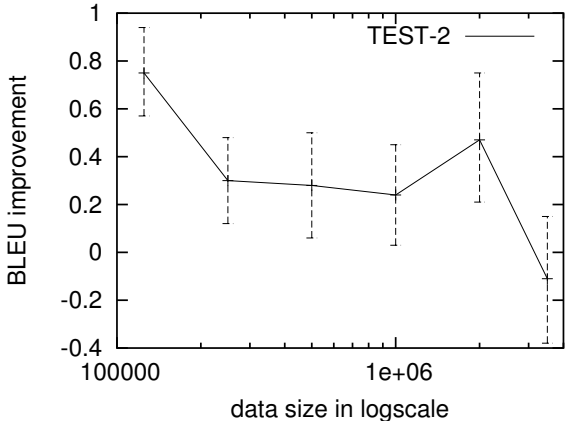

Figure 7: BLEU Improvement with 95% confidence range by using factorized grammars on TEST-2.

6.3 Example Templates

Figure 8 lists seven Arabic-to-English templates randomly selected from the transitive verb family. TMPL_151 is an interesting one. It helps to alleviate the pronoun dropping problem in Arabic. However, we notice that most of the templates in the 200 lists are rather simple. More sophisticated solutions are needed to go deep into the list to find out better templates in future.

It will be interesting to find an automatic or semi-automatic way to discover source counterparts of target treelets in the XTAG English Grammar. Generic rules like this will be very close to handcraft translate rules that people have accumulated for rule-based MT systems.

7 Conclusions and Future Work

In this paper, we proposed a novel statistical machine translation model using a factorized structure-based translation grammar. This model not only alleviates the sparse data problem but only relieves the burden on space and search, both of which are imminent issues for the popular phrasal and/or hierarchical MT systems.

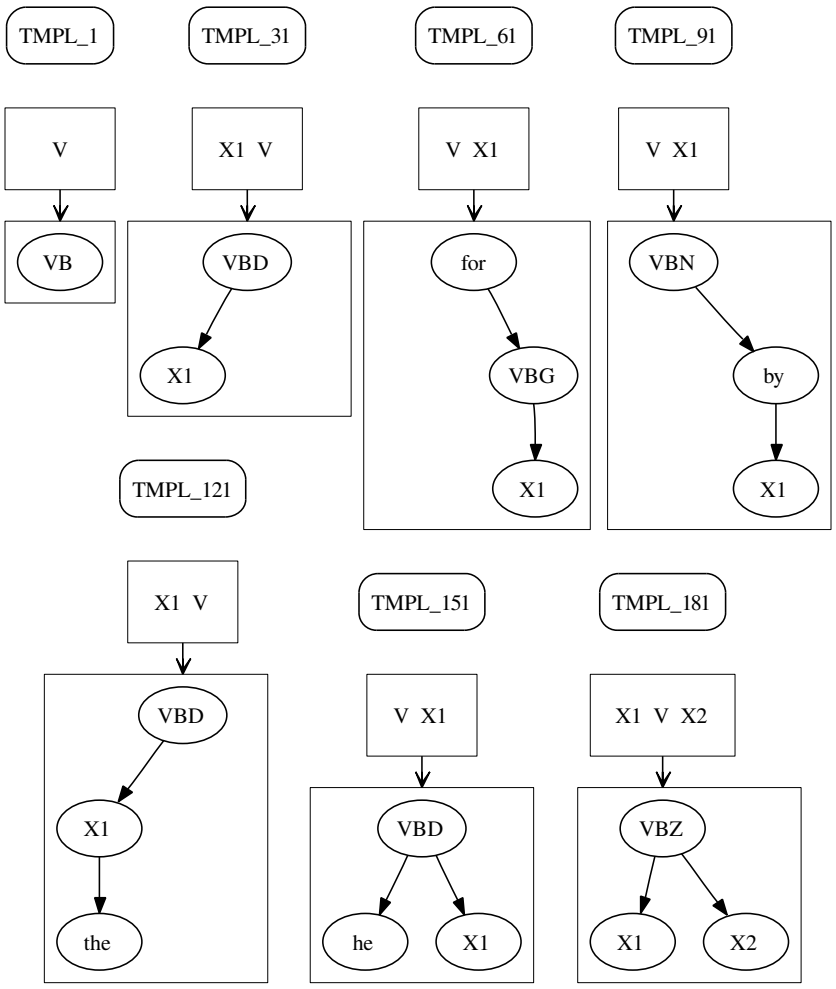

Figure 8: Randomly selected Arabic-to-English templates from the transitive verb family.

We took low-resource language translation, especially X-to-English translation tasks, for case study. We designed a method to exploit family information in the XTAG English Grammar to facilitate the extraction of factorized rules. We tested the new model on low-resource translation, and the use of factorized models showed significant improvement in BLEU on systems with 200K words of bi-lingual training data of various language pairs and genres.

The factorized translation grammar proposed here shows an interesting way of using richer syntactic resources, with high potential for future research.

In future, we will explore various learning methods for better estimation of families, templates and lexical items. The target linguistic knowledge that we used in this paper will provide a nice starting point for unsupervised learning algorithms.

We will also try to further exploit the factorized representation with discriminative learning. Features defined on templates and families will have good generalization capability.

Acknowledgments

This work was supported by DARPA/IPTO Contract HR0011-06-C-0022 under the GALE program[3]. We thank Aravind Joshi, Scott Miller, Richard Schwartz and anonymous reviewers for valuable comments.

[3]Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). 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 official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 101–104, Ann Arbor, MI.

Xavier Carreras and Michael Collins. 2009. Non-projective parsing for statistical machine translation. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing*, pages 200–209, Singapore.

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.

Jacob Devlin. 2009. Lexical features for statistical machine translation. Master's thesis, Univ. of Maryland.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. The MIT Press.

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.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extensive classifications of english verbs. In *Proceedings of the 12th EURALEX International Congress*.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Conference of Empirical Methods in Natural Language Processing*.

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.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference of Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

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.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

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.

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)*.

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. 2009. Effective Use of Linguistic and Contextual Information for Statistical Machine Translation. In *Proceedings of the 2009 Conference of Empirical Methods in Natural Language Processing*, pages 72–80, Singapore.

XTAG-Group. 2001. A lexicalized tree adjoining grammar for english. Technical Report 01-03, IRCS, Univ. of Pennsylvania.