Syntactic SMT Using a Discriminative Text Generation Model

Yue Zhang
SUTD, Singapore
yue_zhang@sutd.edu.sg

Kai Song*
NEU, China
songkai.sk@alibaba-inc.com

Linfeng Song*
ICT/CAS, China
songlinfeng@ict.ac.cn

Jingbo Zhu
NEU, China
zhujingbo@mail.neu.edu.cn

Qun Liu
CNGL, Ireland and ICT/CAS, China
qliu@computing.dcu.ie

Abstract

We study a novel architecture for syntactic SMT. In contrast to the dominant approach in the literature, the system does not rely on translation rules, but treat translation as an unconstrained target sentence generation task, using soft features to capture lexical and syntactic correspondences between the source and target languages. Target syntax features and bilingual translation features are trained consistently in a discriminative model. Experiments using the IWSLT 2010 dataset show that the system achieves BLEU comparable to the state-of-the-art syntactic SMT systems.

1 Introduction

Translation rules have been central to hierarchical phrase-based and syntactic statistical machine translation (SMT) (Galley et al., 2004; Chiang, 2005; Liu et al., 2006; Quirk et al., 2005; Marcu et al., 2006; Shen and Joshi, 2008; Xie et al., 2011). They are attractive by capturing the recursiveness of languages and syntactic correspondences between them. One important advantage of translation rules is that they allow efficient decoding by treating MT as a statistical **parsing** task, transforming a source sentence to its translation via recursive rule application.

The efficiency takes root in the fact that target word orders are encoded in translation rules. This fact, however, also leads to rule explosion, noise and coverage problems (Auli et al., 2009), which can hurt translation quality. Flexibility of function word usage, rich morphology and paraphrasing all add to the difficulty of rule extraction. In addition, restricting target word orders by hard translation rules can also hurt output fluency.

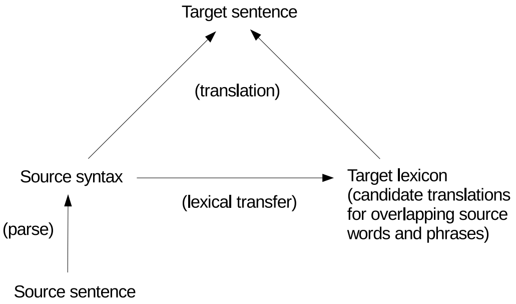

Figure 1: Overall system architecture.

A potential solution to the problems above is to treat translation as a **generation** task, representing syntactic correspondences using *soft* features. Both adequacy and fluency can potentially be improved by giving full flexibility to target synthesis, and leaving all options to the statistical model. The main challenge to this method is a significant increase in the search space (Knight, 1999). To this end, recent advances in tackling complex search tasks for text generation offer some solutions (White and Rajkumar, 2009; Zhang and Clark, 2011).

In this short paper, we present a preliminary investigation on the possibility of building a syntactic SMT system that does not use hard translation rules, by utilizing recent advances in statistical natural language generation (NLG). The overall architecture is shown in Figure 1. Translation is performed by first parsing the source sentence, then transferring source words and phrases to their target equivalences, and finally synthesizing the target output.

We choose dependency grammar for both the source and the target syntax, and adapt the syntactic text synthesis system of Zhang (2013), which performs dependency-based linearization. The linearization task for MT is different from the monolingual task in that not all translation options are used to build the output, and that bilingual correspondences need to be taken into account dur-

* Work done while visiting Singapore University of Technology and Design (SUTD)

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 177–182,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

ing synthesis. The algorithms of Zhang (2013) are modified to perform word selection as well as ordering, using two sets of features to control translation adequacy and fluency, respectively.

Preliminary experiments on the IWSLT[1] 2010 data show that the system gives BLEU comparable to traditional tree-to-string and string-to-tree translation systems. It demonstrates the feasibility of leveraging statistical NLG techniques for SMT, and the possibility of building a statistical transfer-based MT system.

2 Approach

The main goal being proof of concept, we keep the system simple by utilizing existing methods for the main components, minimizing engineering efforts. Shown in Figure 1, the end-to-end system consists of two main components: *lexical transfer* and *synthesis*. The former provides candidate translations for (overlapping) source words and phrases. Although lexicons and rules can be used for this step, we take a simple statistical alignment-based approach. The latter searches for a target translation by constructing dependency trees bottom-up. The process can be viewed as a syntax-based generation process from a bag of overlapping translation options.

2.1 Lexical transfer

We perform word alignment using IBM model 4 (Brown et al., 1993), and then extract phrase pairs according to the alignment and automatically-annotated target syntax. In particular, consistent (Och et al., 1999) and cohesive (Fox, 2002) phrase pairs are extracted from intersected alignments in both directions: the target side must form a projective span, with a single root, and the source side must be contiguous. A resulting phrase pair consists of the source phrase, its target translation, as well as the head position and head part-of-speech (POS) of the target span, which are useful for target synthesis. We further restrict that neither the source nor the target side of a valid phrase pair contains over s words.

Given an input source sentence, the lexical transfer unit finds all valid target translation options for overlapping source phrases up to size s, and feeds them as inputs to the target synthesis decoder. The translation options with a probability below $\lambda \cdot P_{max}$ are filtered out, where P_{max} is the probability of the most probable translation. Here the probability of a target translation is calculated as the count of the translation divided by the count of all translations of the source phrase.

2.2 Synthesis

The synthesis module is based on the monolingual text synthesis algorithm of Zhang (2013), which constructs an ordered dependency tree given a bag of words. In the bilingual setting, inputs to the algorithm are translation options, which can be overlapping and mutually exclusive, and not necessarily all of which are included in the output. As a result, the decoder needs to perform word selection in addition to word ordering. Another difference between the bilingual and monolingual settings is that the former requires translation adequacy in addition to output fluency.

We largely rely on the monolingual system for MT decoding. To deal with overlapping translation options, a source coverage vector is used to impose mutual exclusiveness on input words and phrases. Each element in the coverage vector is a binary value that indicates whether a particular source word has been translated in the corresponding target hypothesis. For translation adequacy, we use a set of bilingual features on top of the set of monolingual features for text synthesis.

2.2.1 Search

The search algorithm is the best-first algorithm of Zhang (2013). Each search hypothesis is a partial or full target-language dependency tree, and hypotheses are constructed bottom-up from leaf nodes, which are translation options. An *agenda* is used to maintain a list of search hypothesis to be expanded, and a *chart* is used to record a set of accepted hypotheses. Initially empty, the chart is a beam of size $k \cdot n$, where n is the number of source words and k is a positive integer. The agenda is a priority queue, initialized with all leaf hypotheses (i.e. translation options). At each step, the highest-scored hypothesis e is popped off the agenda, and expanded by combination with all hypotheses on the chart in all possible ways, with the set of newly generated hypotheses $e_1, e_2, ...e_N$ being put onto the agenda, and e being put onto the chart. When two hypotheses are combined, they can be put in two different orders, and in each case different dependencies can be constructed between their head words, leading to different new

[1] International Workshop on Spoken Language Translation, *http://iwslt2010.fbk.eu*

dependency syntax
WORD(h) · POS(h) · NORM($size$) , WORD(h) · NORM($size$), POS(h) · NORM($size$)
POS(h) · POS(m) · POS(b) · dir
POS(h) · POS(h_l) · POS(m) · POS(m_r) · dir ($h > m$), POS(h) · POS(h_r) · POS(m) · POS(m_l) · dir ($h < m$)
WORD(h) · POS(m) · POS(m_l) · dir, WORD(h) · POS(m) · POS(m_r) · dir
POS(h) · POS(m) · POS(m_1) · dir, POS(h) · POS(m_1) · dir, POS(m) · POS(m_1) · dir
WORD(h) · POS(m) · POS(m_1) · POS(m_2) · dir, POS(h) · POS(m) · POS(m_1) · POS(m_2) · dir , ...
dependency syntax for completed words
WORD(h) · POS(h) · WORD(h_l) · POS(h_l), POS(h) · POS(h_l), WORD(h) · POS(h) · POS(h_l), POS(h) · WORD(h_l) · POS(h_l) , WORD(h) · POS(h) · WORD(h_r) · POS(h_r), POS(h) · POS(h_r), ...
surface string patterns (B—bordering index)
WORD($B-1$) · WORD(B), POS($B-1$) · POS(B), WORD($B-1$) · POS(B), POS($B-1$) · WORD(B), WORD($B-1$) · WORD(B) · WORD($B+1$), WORD($B-2$) · WORD($B-1$) · WORD(B), POS($B-1$) · POS(B) · POS($B+1$), ...
surface string patterns for complete sentences
WORD(0), WORD(0) · WORD(1), WORD($size-1$), WORD($size-1$) · WORD($size-2$), POS(0), POS(0) · POS(1), POS(0) · POS(1) · POS(2), ...

Table 1: Monolingual feature templates.

phrase translation features
PHRASE(m) · PHRASE(t), $P(trans)$,
bilingual syntactic features
POS(th) · POS(tm) · dir · LEN($path$), WORD(th) · POS(tm) · dir · LEN($path$), POS(th) · WORD(tm) · dir · LEN($path$), WORD(th) · WORD(tm) · dir · LEN($path$), WORD(sh) · WORD(sm) · dir · LEN($path$), WORD(sh) · WORD(th) · dir · LEN($path$), WORD(sm) · WORD(tm) · dir · LEN($path$),
bilingual syntactic features (LEN($path$) $\leq$ 3)
POS(th) · POS(tm) · dir · LABELS($path$), WORD(th) · POS(tm) · dir · LABELS($path$), POS(th) · WORD(tm) · dir · LABELS($path$), WORD(th) · WORD(tm) · dir · LABELS($path$), WORD(sh) · WORD(sm) · dir · LABELS($path$), WORD(sh) · WORD(th) · dir · LABELS($path$), WORD(sm) · WORD(tm) · dir · LABELS($path$), POS(th) · POS(tm) · dir · LABELSPOS($path$), WORD(th) · POS(tm) · dir · LABELSPOS($path$), POS(th) · WORD(tm) · dir · LABELSPOS($path$), WORD(th) · WORD(tm) · dir · LABELSPOS($path$), WORD(sh) · WORD(sm) · dir · LABELSPOS($path$), WORD(sh) · WORD(th) · dir · LABELSPOS($path$), WORD(sm) · WORD(tm) · dir · LABELSPOS($path$),

Table 2: Bilingual feature templates.

hypotheses. The decoder expands a fixed number L hypotheses, and then takes the highest-scored chart hypothesis that contains over $\beta \cdot n$ words as the output, where β is a real number near 1.0.

2.2.2 Model and training

A scaled linear model is used by the decoder to score search hypotheses:

$$Score(e) = \frac{\vec{\theta} \cdot \Phi(e)}{|e|},$$

where $\Phi(e)$ is the global feature vector of the hypothesis e, $\vec{\theta}$ is the parameter vector of the model, and $|e|$ is the number of leaf nodes in e. The scaling factor $|e|$ is necessary because hypotheses with different numbers of words are compared with each other in the search process to capture translation equivalence.

While the monolingual features of Zhang (2013) are applied (example feature templates from the system are shown in Table 1), an additional set of bilingual features is defined, shown in Table 2. In the tables, s and t represent the source and target, respectively; h and m represent the head and modifier in a dependency arc, respectively; h_l and h_r represent the neighboring words on the left and right of h, respectively; m_l and m_r represent the neighboring words on the left and right of m, respectively; m_1 and m_2 represent the closest and second closest sibling of m on the side of h, respectively. dir represents the arc direction (i.e. left or right); PHRASE represents a lexical phrase; P($trans$) represents the source-to-target translation probability from the phrase-table, used as a real-valued feature; $path$ represents the shortest path in the source dependency tree between the two nodes that correspond to the target head and modifier, respectively; LEN($path$) represents the number of arcs on *path*, normalized to bins of [5, 10, 20, 40+]; LABELS($path$) represents the array of dependency arc labels on *path*; LABELSPOS($path$) represents the array of dependency arc labels and source POS on *path*. In addition, a real-valued four-gram language model feature is also used, with four-grams extracted from the surface boundary when two hypothesis are combined.

We apply the discriminative learning algorithm of Zhang (2013) to train the parameters $\vec{\theta}$. The algorithm requires training examples that consist of full target derivations, with leaf nodes being *input translation options*. However, the readily available training examples are automatically-parsed target derivations, with leaf nodes being *the reference translation*. As a result, we apply a search procedure to find a derivation process, through which the target dependency tree is constructed from a subset of input translation options. The search procedure can be treated as a constrained decoding process, where only the oracle tree and its sub trees can be constructed. In case the set of translation options cannot lead to the oracle tree, we ignore the training instance.[2] Although the ignored training sentence pairs cannot be utilized for training the discriminative synthesizer, they are nevertheless used for building the phrase table and training the language model.

3 Experiments

We perform experiments on the IWSLT 2010 Chinese-English dataset, which consists of training sentence pairs from the dialog task (dialog) and Basic Travel and Expression Corpus (BTEC). The union of dialog and BTEC are taken as our training set, which contains 30,033 sentence pairs. For system tuning, we use the IWSLT 2004 test set (also released as the second development test set of IWSLT 2010), which contains 500 sentences. For final test, we use the IWSLT 2003 test set (also released as the first development test set of IWSLT 2010), which contains 506 sentences.

The Chinese sentences in the datasets are segmented using NiuTrans[3] (Xiao et al., 2012), while POS-tagging of both English and Chinese is performed using ZPar[4] version 0.5 (Zhang and Clark, 2011). We train the English POS-tagger using the WSJ sections of the Penn Treebank (Marcus et al., 1993), turned into lower-case. For syntactic parsing of both English and Chinese, we use the default models of ZPar 0.5.

We choose three baseline systems: a string-to-tree (S2T) system, a tree-to-string (T2S) system and a tree-to-tree (T2T) system (Koehn, 2010). The Moses release 1.0 implementations of all three systems are used, with default parameter settings. IRSTLM[5] release 5.80.03 (Federico et al., 2008) is used to train a four-gram language models over the English training data, which is applied to the baseline systems and our system. Kneser-Ney smoothing is used to train the language model.

System	T2S	S2T	T2T	OURS
BLEU	32.65	36.07	28.46	34.24

Table 3: Final results.

SOURCE: 我 现在 头痛 的 厉害 。
REF: I have a terrible headache .
OURS: now , I have a headache .
SOURCE: 我 要 带 浴缸 的 双人房 。
REF: I ’d like a twin room with a bath please .
OURS: a twin room , I ’ll find a room with a bath .
SOURCE: 请 把 日元 兑换 成 美元 。
REF: can you change yen into dollars ?
OURS: please change yen into dollars .
SOURCE: 请 给 我 烤鸡 。
REF: roast chicken , please .
OURS: please have roast chicken .
SOURCE: 请 每 次 饭 后 吃 两 粒 。
REF: take two tablets after every meal .
OURS: please eat after each meal .
SOURCE: 请 结帐 。
REF: check , please .
OURS: I have to check - out , please .
SOURCE: 对 呀 那 是 本店 最 拿手 的 菜 啊 。
REF: yes , well , that ’s our specialty .
OURS: ah , the food that ’s right .
SOURCE: 空调 坏 了 。
REF: my air conditioner is n’t working .
OURS: the air - conditioner does n’t work .

Table 4: Sample output sentences.

We use the tuning set to determine the optimal number of training iterations. The translation option filter λ is set to 0.1; the phrase size limit s is set to 5 in order to verify the effectiveness of synthesis; the number of expanded nodes L is set to 200; the chart factor k is set to 16 for a balance between efficiency and accuracy; the goal parameter β is set to 0.8.

The final scores of our system and the baselines are shown in Table 3. Our system gives a BLEU of 34.24, which is comparable to the baseline systems. Some example outputs are shown in Table 4. Manual comparison does not show significant differences in overall translation adequacy or fluency between the outputs of the four systems. However, an observation is that, while our system can produce more fluent outputs, the choice of translation options can be more frequently incorrect. This suggests that while the target synthesis component is effective under the bilingual setting, a stronger lexical selection component may be necessary for better translation quality.

[2] This led to the ignoring of over 40% of the training sentence pairs. For future work, we will consider substitute oracles from reachable target derivations by using maximum sentence level BLEU approximation (Nakov et al., 2012) or METEOR (Denkowski and Lavie, 2011) as selection criteria.

[3] *http://www.nlplab.com/NiuPlan/NiuTrans.ch.html*

[4] *http://sourceforge.net/projects/zpar/*

[5] *http://sourceforge.net/apps/mediawiki/irstlm*

4 Related work

As discussed in the introduction, our work is closely related to previous studies on syntactic MT, with the salient difference that we do not rely on hard translation rules, but allow free target synthesis. The contrast can be summarized as "translation by parsing" vs "translation by generation".

There has been a line of research on generation for translation. Soricut and Marcu (2006) use a form of weighted IDL-expressions (Nederhof and Satta, 2004) for generation. Bangalore et al. (2007) treats MT as a combination of global lexical transfer and word ordering; their generation component does not perform lexical selection, relying on an n-gram language model to order target words. Goto et al. (2012) use a monotonic phrase-based system to perform target word selection, and treats target ordering as a post-processing step. More recently, Chen et al. (2014) translate source dependencies arc-by-arc to generate pseudo target dependencies, and generate the translation by reordering of arcs. In contrast with these systems, our system relies more heavily on a syntax-based synthesis component, in order to study the usefulness of statistical NLG on SMT.

With respect to syntax-based word ordering, Chang and Toutanova (2007) and He et al. (2009) study a simplified word ordering problem by assuming that the un-ordered target dependency tree is given. Wan et al. (2009) and Zhang and Clark (2011) study the ordering of a bag of words, without input syntax. Zhang et al. (2012), Zhang (2013) and Song et al. (2014) further extended this line of research by adding input syntax and allowing joint inflection and ordering. de Gispert et al. (2014) use a phrase-structure grammer for word ordering. Our generation system is based on the work of Zhang (2013), but further allows lexical selection.

Our work is also in line with the work of Liang et al. (2006), Blunsom et al. (2008), Flanigan et al. (2013) and Yu et al. (2013) in that we build a discriminative model for SMT.

5 Conclusion

We investigated a novel system for syntactic machine translation, treating MT as an unconstrained generation task, solved by using a single discriminative model with both monolingual syntax and bilingual translation features. Syntactic correspondence is captured by using soft features rather than hard translation rules, which are used by most syntax-based statistical methods in the literature.

Our results are preliminary in the sense that the experiments were performed using a relatively small dataset, and little engineering effort was made on fine-tuning of parameters for the baseline and proposed models. Our Python implementation gives the same level of BLEU scores compared with baseline syntactic SMT systems, but is an order of magnitude slower than Moses. However, the results demonstrate the feasibility of leveraging text generation techniques for machine translation, directly connecting the two currently rather separated research fields. The system is not strongly dependent on the specific generation algorithm, and one potential of the SMT architecture is that it can directly benefit from advances in statistical NLG technology.

Acknowledgement

The work has been supported by the Singapore Ministration of Education Tier 2 project T2MOE201301 and the startup grant SRG ISTD 2012 038 from SUTD. We thank the anonymous reviewers for their constructive comments.

References

Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. A systematic analysis of translation model search spaces. In *Proc. WMT*, pages 224–232.

Srinivas Bangalore, Patrick Haffner, and Stephan Kanthak. 2007. Statistical machine translation through global lexical selection and sentence reconstruction. In *Proc. ACL*, pages 152–159.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. ACL*, pages 200–208.

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

Pi-Chuan Chang and Kristina Toutanova. 2007. A discriminative syntactic word order model for machine translation. In *Proc. ACL*, pages 9–16.

Hongshen Chen, Jun Xie, Fandong Meng, Wenbin Jiang, and Qun Liu. 2014. A dependency edge-based transfer model for statistical machine translation. In *Proc. COLING 2014*, pages 1103–1113.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL*, pages 263–270.

Adrià de Gispert, Marcus Tomalin, and Bill Byrne. 2014. Word ordering with phrase-based grammars. In *Proc. EACL*, pages 259–268.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proc. WMT*, pages 85–91.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proc. Interspeech*, pages 1618–1621.

Jeffrey Flanigan, Chris Dyer, and Jaime Carbonell. 2013. Large-scale discriminative training for statistical machine translation using held-out line search. In *Proc. NAACL*, pages 248–258.

Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proc. EMNLP*, pages 304–311.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. HLT-NAACL*, pages 273–280.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by parsing for Japanese-English statistical machine translation. In *Proc. ACL*, pages 311–316.

Wei He, Haifeng Wang, Yuqing Guo, and Ting Liu. 2009. Dependency based Chinese sentence realization. In *Proc. ACL/AFNLP*, pages 809–816.

Kevin Knight. 1999. Squibs and Discussions: Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, 25(4):607–615.

Phillip Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

P. Liang, A. Bouchard-Cote, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. COLING/ACL*, pages 761–768.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. COLING/ACL*, pages 609–616.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. EMNLP*, pages 44–52.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The penn treebank. *Computational linguistics*, 19(2):313–330.

Preslav Nakov, Francisco Guzman, and Stephan Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *Proc. Coling*, pages 1979–1994.

Mark-Jan Nederhof and Giorgio Satta. 2004. Idl-expressions: a formalism for representing and parsing finite languages in natural language processing. *J. Artif. Intell. Res.(JAIR)*, 21:287–317.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. EMNLP*, pages 20–28.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proc. ACL*, pages 271–279.

Libin Shen and Aravind Joshi. 2008. LTAG dependency parsing with bidirectional incremental construction. In *Proc. EMNLP*, pages 495–504.

Linfeng Song, Yue Zhang, Kai Song, and Qun Liu. 2014. Joint morphological generation and syntactic linearization. In *Proc. AAAI*, pages 1522–1528.

Radu Soricut and Daniel Marcu. 2006. Stochastic language generation using widl-expressions and its application in machine translation and summarization. In *Proc. ACL*, pages 1105–1112.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2009. Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In *Proc. EACL*, pages 852–860.

Michael White and Rajakrishnan Rajkumar. 2009. Perceptron reranking for CCG realization. In *Proc. the EMNLP*, pages 410–419.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. NiuTrans: An open source toolkit for phrase-based and syntax-based machine translation. In *Proc. ACL Demos*, pages 19–24.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proc. EMNLP*, pages 216–226.

Heng Yu, Liang Huang, Haitao Mi, and Kai Zhao. 2013. Max-violation perceptron and forced decoding for scalable MT training. In *Proc. EMNLP*, pages 1112–1123.

Yue Zhang and Stephen Clark. 2011. Syntax-based grammaticality improvement using CCG and guided search. In *Proc. EMNLP*, pages 1147–1157.

Yue Zhang, Graeme Blackwood, and Stephen Clark. 2012. Syntax-based word ordering incorporating a large-scale language model. In *Proc. EACL*, pages 736–746.

Yue Zhang. 2013. Partial-tree linearization: Generalized word ordering for text synthesis. In *Proc. IJCAI*, pages 2232–2238.