Using a Supertagged Dependency Language Model to Select a Good Translation in System Combination

Wei-Yun Ma
Department of Computer Science
Columbia University
New York, NY 10027, USA
ma@cs.columbia.edu

Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027, USA
kathy@cs.columbia.edu

Abstract

We present a novel, structured language model - Supertagged Dependency Language Model to model the syntactic dependencies between words. The goal is to identify ungrammatical hypotheses from a set of candidate translations in a MT system combination framework and help select the best translation candidates using a variety of sentence-level features. We use a two-step mechanism based on constituent parsing and elementary tree extraction to obtain supertags and their dependency relations. Our experiments show that the structured language model provides significant improvement in the framework of sentence-level system combination.

1 Introduction

In recent years, there has been a burgeoning interest in incorporating syntactic structure into Statistical machine translation (SMT) models (e..g, Galley et al., 2006; DeNeefe and Knight 2009; Quirk et al., 2005). In addition to modeling syntactic structure in the decoding process, a methodology for candidate translation selection has also emerged. This methodology first generates multiple candidate translations followed by rescoring using global sentence-level syntactic features to select the final translation. The advantage of this methodology is that it allows for easy integration of complex syntactic features that would be too expensive to use during the decoding process. The methodology is usually applied in two scenarios: one is as part of an n-best reranking (Och et al., 2004; Hasan et al., 2006), where n-best candidate translations are generated through a decoding process. The other is translation selection or reranking (Hildebrand and Vogel 2008; Callison-Burch et al., 2012), where candidate translations are generated by different decoding processes or different decoders.

This paper belongs to the latter; the goal is to identify ungrammatical hypotheses from given candidate translations using grammatical knowledge in the target language that expresses syntactic dependencies between words. To achieve that, we propose a novel Structured Language Model (SLM) - Supertagged Dependency Language Model (SDLM) to model the syntactic dependencies between words. Supertag (Bangalore and Joshi, 1999) is an elementary syntactic structure based on Lexicalized Tree Adjoining Grammar (LTAG). Traditional supertagged n-gram LM predicts the next supertag based on the immediate words to the left with supertags, so it can not explicitly model long-distance dependency relations. In contrast, SDLM predicts the next supertag using the words with supertags on which it syntactically depend, and these words could be anywhere and arbitrarily far apart in a sentence. A candidate translation's grammatical degree or "fluency" can be measured by simply calculating the SDLM likelihood of the supertagged dependency structure that spans the entire sentence.

To obtain the supertagged dependency structure, the most intuitive way is through a LTAG parser (Schabes et al., 1988). However, this could be very

Proceedings of NAACL-HLT 2013, pages 433–438,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

slow as it has time complexity of O(n^6). Instead we propose an alternative mechanism in this paper: first we use a constituent parser[1] of O(n^3) ~ O(n^5) to obtain the parse of a sentence, and then we extract elementary trees with dependencies from the parse in linear time. Aside from the consideration of time complexity, another motivation of this two-step mechanism is that compared with LTAG parsing, the mechanism is more flexible for defining syntactic structures of elementary trees for our needs. Because those structures are defined only within the elementary tree extractor, we can easily adjust the definition of those structures within the extractor and avoid redesigning or retraining our constituent parser.

We experiment with sentence-level translation combination of five different translation systems; the goal is for the system to select the best translation for each input source sentence among the translations provided by the five systems. The results show a significant improvement of 1.45 Bleu score over the best single MT system and 0.72 Bleu score over a baseline sentence-level combination system of using consensus and n-gram LM.

2 Related Work

Och et al., (2004) investigated various syntactic feature functions to rerank the n-best candidate translations. Most features are syntactically motivated and based on alignment information between the source sentence and the target translation. The results are rather disappointing. Only the non-syntactic IBM model 1 yielded significant improvement. All other tree-based feature functions had only a very small effect on the performance.

In contrast to (Och et al., 2004)'s bilingual syntax features, Hasan et al., (2006) focused on monolingual syntax features in n-best reranking. They also investigated the effect of directly using the log-likelihood of the output of a HMM-based supertagger, and found it did not improve performance significantly. It is worth noticing that this log-likelihood is based on supertagged n-gram LM, which is one type of class-based n-gram LM, so it does not model explicit syntactic dependencies between words in contrast to the work we describe in this paper. Hardmeier et al., (2012) use tree kernels over constituency and dependency parse trees for either the input or output sentences to identify constructions that are difficult to translate in the source language, and doubtful syntactic structures in the output language. The tree fragments extracted by their tree kernels are similar to our elementary trees but they only regard them as the individual inputs of support vector machine regression while binary relations of our elementary trees are considered in a formulation of a structural language model.

Outside the field of candidate translation selection, Hassan et al., (2007) proposed a phrase-based SMT model that integrates supertags into the target side of the translation model and the target n-gram LM. Two kinds of supertags are employed: those from LTAG and Combinatory Categorial Grannar (CCG), and both yield similar improvements. They found that using both or either of the supertag-based translation model and supertagged LM can achieve significant improvement. Again, the supertagged LM is a class-based n-gram LM and does not model explicit syntactic dependencies during decoding.

In the field of MT system combination, word-level confusion network decoding is one of the most successful approaches (Matusov et al., 2006; Rosti et al., 2007; He et al. 2008; Karakos et al. 2008; Sim et al. 2007; Xu et al. 2011). It is capable of generating brand new translations but it is difficult to consider more complex syntax such as dependency LM during decoding since it adds one word at a time while a dependency based LM must parse a complete sentence. Typically, a confusion network approach selects one translation as the best and uses this as the backbone for the confusion network. The work we present here could provide a more sophisticated mechanism for selecting the backbone. Alternatively, one can enhance confusion network models by collaborating with a sentence-level combination model which uses complex syntax to re-rank n-best outputs of a confusion network model. This kind of collaboration is one of our future works.

[1] Stanford parser (http://nlp.stanford.edu/software/lex-parser.shtml). We use its PCFG version of O(n^3) for SDLM training of part of Gigaword in addition to Treebank and use its factor version of O(n^5) to calculate the SDLM likelihood of translations.

3 LTAG and Supertag

LTAG (Joshi et al., 1975; Schabes et al., 1988) is a formal tree rewriting formalism, which consists of a set of elementary trees, corresponding to minimal linguistic structures that localize dependencies, including long-distance dependencies, such as predicate-argument structure. Each elementary tree is associated with at least one lexical item on its frontier. The lexical item associated with an elementary tree is called the anchor in that tree; an elementary tree thus serves as a description of syntactic constraints of the anchor. The elementary syntactic structures of elementary trees are called supertags (Bangalore and Joshi, 1999), in order to distinguish them from the standard part-of-speech tags. Some examples are provided in figure 1 (b).

Elementary trees are divided into initial and auxiliary trees. Initial trees are those for which all non-terminal nodes on the frontier are substitutable. Auxiliary trees are defined as initial trees, except that exactly one frontier, non-terminal node must be a foot node, with the same label as the root node. Two operations - substitution and adjunction - are provided in LTAG to combine elementary trees into a derived tree.

4 SDLM

Our goal is to use SDLM to calculate the grammaticality of translated sentences. We do this by calculating the likelihood of the supertagged dependency structure that spans the entire sentence using SDLM. To obtain the supertagged dependency linkage, the most intuitive way is through a LTAG parser (Schabes et al., 1988). However, this could be very slow as it has time complexity of $O(n^6)$. Another possibility is to follow the procedure in (Joshi and Srinivas 1994, Bangalore and Joshi, 1999): use a HMM-based supertagger to assign words with supertags, followed by derivation of a shallow parse in linear time based on only the supertags to obtain the dependencies. But since this approach uses only the local context, in (Joshi and Srinivas 1994), they also proposed another greedy algorithm based on supertagged dependency probabilities to gradually select the path with the maximum path probability to extend to the remaining directions in the dependency list.

In contrast to the LTAG parsing and supertagging-based approaches, we propose an alternative mechanism: first we use a state-of-the-art constituent parser to obtain the parse of a sentence, and then we extract elementary trees with dependencies from the parse to assign each word with an elementary tree. The second step is similar to the approach used in extracting elementary trees from the TreeBank (Xia, 1999; Chen and Vijay-Shanker, 2000).

4.1 Elementary Tree Extraction

We use an elementary tree extractor, a modification of (Chen and Vijay-Shanker, 2000), to serve our purpose. Heuristic rules were used to distinguish arguments from adjuncts, and the extraction process can be regarded as a process that gradually decomposes a constituent parse to multiple elementary trees and records substitutions and adjunctions. From elementary trees, we can obtain supertags by only considering syntactic structure and ignoring anchor words. Take the sentence – “The hungry boys ate dinner” as an example; the constituent parse and extracted supertags are shown in Figure 1.

In Figure 1 (b), dotted lines represent the operations of substitution and adjunction. Note that each word in a translated sentence would be assigned exactly one elementary syntactic structure which is associated with a unique supertag id for the whole corpus. Different anchor words could own the same elementary syntactic structure and would be assigned the same supertag id, such as “ $\alpha 1$ ” for “boys” and “dinner”. For our corpus, around 1700 different elementary syntactic structures (1700 supertag ids) are extracted.

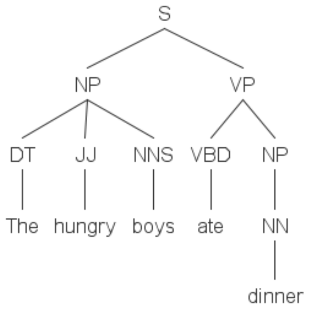

Figure 1. (a) Parse of “The hungry boys ate dinner”

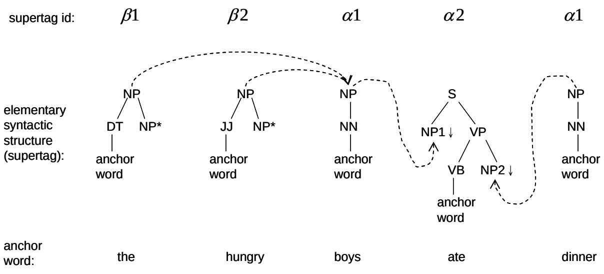

Figure 1. (b) Extracted elementary trees

4.2 Model

Bangalore and Joshi (1999) gave a concise description for dependencies between supertags: "A supertag is dependent on another supertag if the former substitutes or adjoins into the latter". Following this description, for the example in Figure 1 (b), supertags of "the" and "hungry" are dependent on the supertag of "boys", and supertags of "boys" and "dinner" are dependent on the supertag of "ate". These dependencies between supertags also provide the dependencies between anchor words.

Since the syntactic constraints for each word in its context are decided and described through its supertag, the likelihood of SDLM for a sentence could also be regarded as the degree of violations of the syntactic constraints on all words in the sentence. Consider a sentence S = w_1 w_2 …w_n with corresponding supertags T = t_1 t_2 …t_n. We use d_i=j to represent the dependency relations for words or supertags. For example, d_3 = 5 means that w_3 depends on w_5 or t_3 depends on t_5. We propose five different bigram SDLM as follows and evaluate their effects in section 5.

$$\prod_i P(w_i t_i \mid w_{d_i} t_{d_i}) \qquad \text{SDLM model(1)}$$

$$\prod_i P(w_i t_i \mid w_{d_i} t_{d_i}) \approx \prod_i P(t_i \mid t_{d_i}) P(w_i \mid t_i) \qquad \text{SDLM model(2)}$$

$$\prod_i P(t_i \mid t_{d_i}) \qquad \text{SDLM model(3)}$$

$$\prod_i P(w_i \mid t_i) \qquad \text{SDLM model(4)}$$

$$\prod_i P(w_i \mid w_{d_i}) \qquad \text{SDLM model(5)}$$

SDLM model (2) is the approximation form of model (1); model (3) and (4) are individual terms of model (2); model (5) models word dependencies based on elementary tree dependencies. The estimation of the probabilities is done using maximum likelihood estimations with Laplace smoothing. Take Figure 1 (b) as an example; if using model (1), the SDLM likelihood of "The hungry boys ate dinner" is

$$P(the,\beta 1 \mid boys,\alpha 1) * P(hungry,\beta 2 \mid boys,\alpha 1) * P(boys,\alpha 1 \mid ate,\alpha 2) * P(dinner,\alpha 1 \mid ate,\alpha 2) * P(ate,\alpha 2 \mid root)$$

In our experiment on sentence-level translation combination, we use a log-linear model to integrate all features including SDLM models. The corresponding weights are trained discriminatively for Bleu score using Minimum Error Rate Training (MERT).

5 Experiment

Our experiments are conducted and reported on the Chinese-English dataset from NIST 2008 (LDC2010T01). It consists of four human reference translations and corresponding machine translations for the NIST Open MT08 test set, which consists of newswire and web data. The test set contains 105 documents with 1312 sentences and output from 23 machine translation systems. Each system provides the top one translation hypothesis for every sentence. We further divide the NIST Open MT08 test set into the tuning set and test set for our experiment of sentence-level translation combination. We divided the 1312 sentences into tuning data of 524 sentences and the test set of 788 sentences. Out of 23 MT systems, we manually select the top five MT systems as our MT systems for our combination experiment.

In terms of SDLM training, since the size of TreeBank-extracted elementary trees is much smaller compared to most practical n-gram LMs trained from the Gigaword corpus, we also extract elementary trees from automatically-generated parses of part of the Gigaword corpus (around one-year newswire of "afp_eng" in Gigaword 4) in addition to TreeBank-extracted elementary trees.

5.1 Feature Functions

For the baseline combination system, we use the following feature functions in the log-linear model to calculate the score of a system translation.

- Sentence consensus based on Translation Edit Ratio (TER)
- Gigaword-trained 3-gram LM and word penalty

For testing SDLM, in additional to all features that the baseline combination system uses, we add single or multiple SDLM models in the log-linear model, and each SDLM model has its own weight.

5.2 Result

From table 1, we can see that the combination of SDLM model 3, 4 and 5 yields the best performance, which is better than the best MT system by Bleu of 1.45, TER of 0.67 and METEOR of 1.25, and also better than the baseline combination system by Bleu of 0.72, TER of 0.25 and METEOR of 0.44. Compared with SDLM model 5, which represents a type of word dependency LM without labels, the results show that adding appropriate syntactic "labels" (here, they are "supertags") on word dependencies brings benefits.

	Bleu	TER	METEOR
Best MT system	30.16	55.45	54.43
baseline	30.89	55.03	55.24
baseline+ model 1	31.29	54.99	55.63
baseline+ model 2	31.25	55.23	55.37
baseline+ model 3	31.25	55.06	55.40
baseline+ model 4	31.44	54.70	55.54
baseline+ model 5	31.39	55.15	55.68
baseline+ model 3+ model 4+ model 5	31.61	54.78	55.68

Table 1. Result of Sentence-level Translation Combination

6 Conclusion

In this paper we presented Supertagged Dependency Language Model for explicitly modeling syntactic dependencies of the words of translated sentences. Our goal is to select the most grammatical translation from candidate translations. To obtain the supertagged dependency structure of a translation candidate, a two-step mechanism based on constituent parsing and elementary tree extraction is also proposed. SDLM shows its effectiveness in the scenario of translation selection.

There are several avenues for future work: we have focused on bigram dependencies in our models; extension to more than two dependent elementary trees is straightforward. It would also be worth investigating the performance of using our sentence-level model to re-rank n-best outputs of a confusion network model. And in terms of applications, SDLM can be directly applied to many other NLP tasks, such as speech recognition and natural language generation.

Acknowledgments

We would like to thank Owen Rambow for providing the elementary tree extractor and also thank the anonymous reviewers for their helpful comments. This work is supported by the National Science Foundation via Grant No. 0910778 entitled "Richer Representations for Machine Translation". All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn treebank. *In Proceedings of the Sixth International Workshop on Parsing Technologies*

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. *In Proceedings of WMT12.*

Steve DeNeefe and Kevin Knight. 2009 Synchronous Tree Adjoining Machine Translation. *In Proceedings of EMNLP*

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*

Christian Hardmeier, Joakim Nivre and Jörg Tiedemann. 2012. Tree Kernels for Machine Translation Quality Estimation. *In Proceedings of WMT12*

S. Hasan, O. Bender, and H. Ney. 2006. Reranking translation hypotheses using structural properties. *In Proceedings of the EACL'06 Workshop on Learning Structured Information in Natural Language Applications*

Hany Hassan , Khalil Sima'an and Andy Way. 2007. Supertagged Phrase-Based Statistical Machine Translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis alignment for computing outputs from machine translation systems. *In Proceedings of EMNLP*

Almut Silja Hildebrand and Stephan Vogel. 2008. Combination of machine translation systems via hypothesis selection from combined n-best lists. *In Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*

Aravind K. Joshi and B. Srinivas. 1994. Disambiguation of super parts of speech (or supertags): Almost parsing. *In Proceedings of the 15th International Conference on Computational Linguistics*

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Science*, 10:136–163.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proceedings of EACL*

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004 A smorgasbord of features for statistical machine translation. *In Proceedings of the Meeting of the North American chapter of the Association for Computational Linguistics*

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using ITG-based alignments. *In Proceedings of ACL-HLT*

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT, *In Proceedings of the Association for Computational Linguistics*

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved word-level system combination for machine translation. *In Proceedings of ACL*

Yves Schabes, Anne Abeille and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. *In Proceedings of the 12th International Conference on Computational Linguistics*

K.C. Sim, W.J. Byrne, M.J.F. Gales, H. Sahbi and P.C. Woodland .2007. Consensus Network Decoding for Statistical Machine Translation System Combination. *In Proceedings of ICASSP*

Fei Xia. 1999. Extracting Tree Adjoining Grammars from Bracketed Corpora. *In Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-1999)*

Daguang Xu, Yuan Cao, Damianos Karakos. 2011. Description of the JHU System Combination Scheme for WMT 2011. *In Proceedings of the Sixth Workshop on Statistical Machine Translation*