Detecting and Correcting Syntactic Errors in Machine Translation Using Feature-Based Lexicalized Tree Adjoining Grammars

Wei-Yun Ma
Department of Computer Science
Columbia University
ma@cs.columbia.edu

Kathleen McKeown
Department of Computer Science
Columbia University
kathy@cs.columbia.edu

Abstract

Statistical machine translation has made tremendous progress over the past ten years. The output of even the best systems, however, is often ungrammatical because of the lack of sufficient linguistic knowledge. Even when systems incorporate syntax in the translation process, syntactic errors still result. To address this issue, we present a novel approach for detecting and correcting ungrammatical translations. In order to simultaneously detect multiple errors and their corresponding words in a formal framework, we use feature-based lexicalized tree adjoining grammars (FB-LTAG) [1]. In FB-LTAG, each lexical item is associated with a syntactic elementary tree, in which each node is associated with a set of feature-value pairs, called Attribute Value Matrices (AVMs). AVMs define the lexical item's syntactic usage. Our syntactic error detection works by checking the AVM values of all lexical items within a sentence using a unification framework. Thus, we use the feature structures in the AVMs to detect the error type and corresponding words. In order to simultaneously detect multiple error types and track their corresponding words, we propose a new unification method which allows the unification procedure to continue when unification fails and also to propagate the failure information to relevant words. We call the modified unification a fail propagation unification. Our approach features: 1) the use of XTAG grammar [2], a rule-based English grammar developed by linguists using the FB-LTAG formalism, 2) the ability to simultaneously detect multiple ungrammatical types and their corresponding words by using FB-LTAG's feature unifications, and 3) the ability to simultaneously correct multiple ungrammatical types based on the detection information.

Grammar checking methods are usually divided into three classes: statistic-based checking [3][4][5][6], rule-based checking [7][8][9] and syntax-based checking [10]. Our approach is a mix of rule-based checking and syntax-based checking: The XTAG English grammar is designed by linguists while the detecting procedure is based on syntactic operations which

dynamically reference the grammar. In our procedure for syntactic error detection, we first decomposes each sentence hypothesis parse tree into elementary trees, followed by associating each elementary tree with AVMs through look-up in the XTAG grammar, and finally reconstruct the original parse tree out of the elementary trees using substitution and adjunction operations along with AVM unifications with fail propagation ability. Once error types and their corresponding words are detected, one is able to correct errors based on a unified consideration of all related words under the same error types. In this paper, we present some simple mechanism to handle part of the detected situations. We use our approach to detect and correct translations of six single statistical machine translation systems. The results show that most of the corrected translations are improved.

References

[1] K. Vijay-Shanker and Aravind K. Joshi. 1988. *Feature structure based tree adjoining grammar*. In Proceedings of COLING-88, pp. 714-719

[2] The XTAG-Group. 2001. *A Lexicalized Tree Adjoining Grammar for English*. Technical Report IRCS 01-03, University of Pennsylvania.

[3] Eric S. Atwell and Stephen Elliot. 1987. *Dealing with Ill-formed English Text*. The Computational Analysis of English, Longman.

[4] Md. Jahangir Alam, Naushad UzZaman, Mumit Khan. 2006. *N-gram based Statistical Grammar Checker for Bangla and English*. In Proceedings of ninth International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh.

[5] Shih-Hung Wu, Chen-Yu Su, Tian-Jian Jiang, Wen-Lian Hsu. 2006. *An Evaluation of Adopting Language Model as the Checker of Preposition Usage*. In Proceedings of ROCLING.

[6] Anta Huang, Tsung-Ting Kuo, Ying-Chun Lai, Shou-De Lin. 2010. *Identifying Correction Rules for Auto Editing*. In Proceedings of ROCLING.

[7] Daniel Naber. 2003. *A Rule-Based Style and Grammar Checker*. Diploma Thesis. University of Bielefeld, Germany.

[8] George E. Heidorn. 2000. *Intelligent writing assistance*. A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text. Marcel Dekker, New York. pp. 181-207.

[9] Sara Stymne and Lars Ahrenberg. 2010. *Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation*. In LREC.

[10] Karen Jensen, George E. Heidorn, Stpehen D. Richardson (Eds.). 1993. *Natural language processing: the PLNLP approach*. Kluwer Academic Publishers