2016-03-05

Text Simplification Using Neural Machine Translation

Tong Wang
University of Massachusetts Boston

Et al.

Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Text Simplification Using Neural Machine Translation

Tong Wang and **Ping Chen** and **John Rochford** and **Jipeng Qiang**
Department of Computer Science, University of Massachusetts Boston, tongwang0001@gmail.com
Department of Computer Engineering, University of Massachusetts Boston
Eunice Kennedy Shriver Center, University of Massachusetts Medical School
Department of Computer Science, Hefei University of Technology

Abstract

Text simplification (TS) is the technique of reducing the lexical, syntactical complexity of text. Existing automatic TS systems can simplify text only by lexical simplification or by manually defined rules. Neural Machine Translation (NMT) is a recently proposed approach for Machine Translation (MT) that is receiving a lot of research interest. In this paper, we regard original English and simplified English as two languages, and apply a NMT model–Recurrent Neural Network (RNN) encoder-decoder on TS to make the neural network to learn text simplification rules by itself. Then we discuss challenges and strategies about how to apply a NMT model to the task of text simplification.

Introduction

Text simplification (TS) aims to simplify the lexical, grammatical, or structural complexity of text while retaining its semantic meaning. It can help various groups of people, including children, non-native speakers, the functionally illiterate, and people with cognitive disabilities, to understand text better. Moreover, TS relates to many Natural Language Processing tasks, such as parsing, machine translation, and summarization (Chandrasekar, Doran, and Srinivas 1996).

Many simplification approaches are used to create simplified text, including substituting, adding, or removing words; and shortening, splitting, dropping, or merging sentences. Different simplification approaches are applied based upon context, length, and syntactic structure of source words and sentences (Petersen and Ostendorf 2007). Usually, multiple simplification approaches work together to simplify text. Research into automatic text simplification has been ongoing for decades. It is generally divided into three systems: lexical simplification, rule-based, and machine translation.

A lexical simplification (LS) system simplifies text mainly by substituting infrequently-used and difficult words with frequently-used and easier words. The general process for lexical simplification includes: identification of difficult words; finding synonyms or similar words by various similarity measures (Glavaš and Štajner 2015); ranking and selecting the best candidate word based on criteria such as language model; and keeping the grammar and syntax of a sentence correct. However, the LS system is not able to simplify a complex syntactic structure.

Rule-based systems use handcrafted rules for syntactic simplification, and substitute difficult words using predefined vocabulary (Siddharthan and Mandya 2014). Through analyzing syntactic structure, a sentence with a particular structure can be transformed into a simple structure. For example, if a long sentence contains "not only" and "but also", it could be split into two sentences. The disadvantages are that this kind of simplification system needs significant human-involvement to manually define rules; and that it is hopeless to write an exhaustive set of rules.

Machine Translation (MT) approaches for TS are showing very good performance. Original English and simplified English can be thought of as two different languages. TS would be the process to translate English to simplified English (Some call it Monolingual Machine Translation) (Zhu, Bernhard, and Gurevych 2010). The English Wikipedia and the Simple English Wikipedia can be used to create a parallel corpus of aligned sentences to train the system.

Neural Machine Translation (NMT) is a newly proposed MT approach. It achieves very impressive results on MT tasks (Cho et al. 2014) (Sutskever, Vinyals, and Le 2014). Instead of operating on small components separately, as with a traditional MT system, the NMT system attempts to build a single large neural network such that every component is tuned based upon training sentence pairs. In this paper, we propose to apply the NMT model to the TS task. To the best of our knowledge, it is the first work to apply Neural Machine Translation to text simplification. Contrary to previous approaches, NMT systems do not rely upon similarity measures or heavily handcrafted rules. The deep neural network can learn all simplification rules by itself.

NMT Model on Text Simplification

This section introduces the RNN encoder-decoder NMT model; explains differences between TS and MT tasks; and discusses challenges and strategies for applying RNN encoder-decoder on text simplification.

RNN Encoder-Decoder Model

We define V_s and V_t as the vocabulary of the source language and the target language; and V_s^* and V_t^* as all sequences (sentences) over V_s and V_t respectively. Given a

Machine Translation	Text Simplfication
Most frequent words (e.g., top 15,000) are used for source and target languages. Every out-of-vocabulary word is replaced with UNK token.	Infrequent words cannot be simply ignored in the TS task. It is important to simplify them.
$\lvert V_s\rvert \approx \lvert V_t\rvert$ and $V_s \cap V_t = \emptyset$	$\lvert V_t\rvert \ll \lvert V_s\rvert$ and $V_t \subset V_s$
Nearly no sharing words in source sentence X and target sentence Y.	Some or even all words in Y may remain the same after simplifying X.
The relation between source sentence and target sentence is usually one to one.	The relation could be one to many (splitting) or many to one (merging).

Table 1: Differences Between MT and TS task

source sentence $X = (x_1, x_2, ..., x_l)$, where $X \in V_s^*$, $x_i \in V_s$, and x_i is the ith word in X, the target (simplified) sentence is $Y = (y_1, y_2, ..., y_{l'})$. l and l' are the length of the sentences. The length is not fixed. Our goal is to build a neural network to model the conditional probability $p(Y|X)$, then train the model to maximize the probability.

RNN is a class of Neural Network in which the connections between internal units may form a directed cycle to exhibit the entire history of previous inputs. The structure of RNN makes it an ideal model to process sequence inputs such as sentences with arbitrary lengths. RNN Encoder-Decoder (Cho et al. 2014) (Sutskever, Vinyals, and Le 2014) is a NMT model that can be jointly trained to maximize the conditional probability of a target sequence given a source sequence $p(Y|X)$. It consists of two RNNs. One encodes the source sentence into a fixed length vector. The other decodes the vector into a target sentence. We intend to apply the RNN Encoder-Decoder model to the TS task.

To make RNN capture short-term and long-term dependencies of a sentence, the type of hidden units should be carefully selected. The hidden units can be Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) (Sutskever, Vinyals, and Le 2014) or other gating units (Cho et al. 2014). The gating hidden units are crucial for RNN to "understand" a sentence, since they are able to carry out information from the previous hidden state, and drop irrelevant information.

Challenges and Strategies for Text Simplification using RNN Encoder-Decoder

We may apply the RNN encoder-decoder model in two ways for text simplification. We will use this model as part of a LS system as an additional feature by scoring the candidate words of a source word. A LS system usually uses an N-gram language model and information contents to score candidate words. Theoretically, the NMT model is able to strengthen the LS system to capture linguistic regularities of whole sentences. Once the training of the RNN encoder-decoder is complete, we can produce the simplified sentence by $\hat{Y} = \mathrm{argmax}_Y\ p(Y|X)$. The candidate words will be selected accordingly to maximize the conditional probability.

We can use this model also to simplify English sentences directly. However, there are differences between the TS task and the MT task, which we list in Table 1. Therefore, a single NMT model is not able to handle different text simplification operations. An option is to design a particular neural network or other type of classifier to detect which operation(e.g., splitting, merging) should be applied first. Then, we could design a different RNN encoder-decoder model to simplify a specific simplification operation. For example, the update gate and the forget gate of hidden units should memorize the long dependency for the merging operation of a long input sentence. The hidden units should focus on the short dependency for the splitting operation.

Future Work

This work is at an early stage. Due to the lack of available aligned sentence pairs, the first step is to collect training data. We plan to do so in two ways: crowdsourcing, or automatic discovery of aligned sentences from Simple English Wikipedia and English Wikipedia. We will then train the RNN encoder-decoder to score candidate words on a lexical simplification system. Our final goal is to build the RNN encoder-decoder model to simplify any English sentence.

References

Chandrasekar, R.; Doran, C.; and Srinivas, B. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, 1041–1044. Association for Computational Linguistics.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Glavaš, G., and Štajner, S. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53th Annual Meeting of the Associatioin for Computational Linguistics (ACL-2015, Short Papers)*, 63.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Petersen, S. E., and Ostendorf, M. 2007. Text simplification for language learners: a corpus analysis. In *SLaTE*, 69–72. Citeseer.

Siddharthan, A., and Mandya, A. 2014. Hybrid text simplification using synchronous dependency grammars with handwritten and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 722–731.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Zhu, Z.; Bernhard, D.; and Gurevych, I. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd international conference on computational linguistics*, 1353–1361. Association for Computational Linguistics.