Spelling Correction and Robust Word Recognition

Emily (Xingwen) Guo

February 14, 2018

Although it seems automatically for human being to learn and utilize language for communication and information exchange, it is not trivial for machines to master this ability. Researchers have been working on natural language processing for decades to boost machines with such capability. Specifically, this presentation will focus on techniques developed over the years on assisting human with spelling correction and robust word recognition.

1. Spelling Correction

Usually spelling errors are divided into two groups [5]: Non-word spelling error and real-word spelling error. While non-word error refers to any edit to the word, which not result in a real word (spell $\rightarrow$ spele), real-word spelling errors are errors that landed in a different valid word (staff $\rightarrow$ stuff). We define the distance between error data and original data as edit distance, and it has shown by research analysis that the majority of spelling error are resulted from a single-letter change [1]. Thus, the assumptions could be made that the error word is normally 1 edit distance away from the error word [1], which are mainly consist of insertions, deletions, substitutions and transposition. To calculate the most possible candidate for a non-word error, we need to build up a probability model of the word w in a context based on n-gram. The idea of noisy channel model is also applied [4], since an error word could be seen as a candidate word being projected to a twisted plane.

2. Robust Word Recognition

Cambridge University effect has found out that human being could easily recognize words with jumbled characters. With this finding, neural network based on character inputs and semi-character inputs are proposed to improve the robustness of error word recognition.

2.1. Character-Aware Neural Network Language Model

Character-Aware Neural Language Model (CharCNN) is using convolutional neural networks to predict word level output based on character input [2]. The convolutional neural networks are used to concatenate character embedding, and the output of this network will be redirect to a long short-term memory (LSTM) recurrent neural network language model (RNN). The RNN is implemented with LSTM [3], since vanilla recurrent neural networks are not capable of learning dependency over certain distance from input data.

2.2. Semi-Character Recurrent Neural Network

The Semi-Character Recurrent Neural Network (ScRNN) model is aimed to mimic how human being deal with the Cambridge effect, and model the eye-movement tracking process [4]. It is implemented with a standard recurrent neural network (RNN) and a memory cell in long short-term memory (LSTM) as well. However, the input layer of ScRNN contains only three parts: the beginning (b), internal (i) and the ending (e) [4]. While the beginning and ending focus on single character, the internal vector represents all remaining characters. Though it is less informative than CharCNN, but it has a relatively better performance on robust word reorganization compared with CharCNN [4], since it focuses on the more important information given at the beginning and the end of a word.

3. Spotlight Question

Are there any other possible mechanism that we could utilize to mimic and boost the performance of machine?

4. Reading Materials

1. Sutskever, Ilya, et al. Generating text with recurrent neural networks. Proceedings of the 28th ICML 2011.

2. Kim, Yoon, et al. Character-Aware Neural Language Models. AAAI. 2016.

3. Schmaltz, Allen, et al. Sentence-level grammatical error identification as sequence-to-sequence correction. 2016.

4. Sakaguchi, Keisuke, et al. "Robsut Wrod Reocginiton via Semi-Character Recurrent Neural Network."AAAI. 2017.

5. Jurafsky, Dan, and James H. Martin. Speech and language processing. Chapter 5, Vol. 3. London:: Pearson, 2014.