Measure Text Fluency

Interim Report

Team Name: Finals
Team number: 12
Ruchitha Jujjuru: 2020101093
Yalaka Surya Teja Reddy: 2020101042
Immadisetty Josh ujwal: 2020102018

Current Work

1. Statistical Approach to Evaluation of Sentence Fluency

1.1 Amelioration 1: evaluating sentence fluency using the product of n-grams’ Conditional Probabilities

$$M(W_1W_2...W_m) = [\sum_{i=1}^{m} \log(P(W_i \mid N - Gram))] / m$$

Here, M is the fluency score of the sentence. P (wi / N-Gram) to denote the Conditional Probability of different n-grams in a sentence

1.2 Amelioration 2: Evaluating sentence fluency by discriminatingly treating strange and familiar n-grams

$$
\begin{array}{l}
M(W_1W_2...W_m) = 1/m \\
\textit{for } i = 1, \ldots m \\
\quad \textit{if } (P(W_i \mid N - Gram) \geq ValForGood) \\
\qquad M(W_1W_2..W_m) = M(W_1W_2..W_m) / P(W_i \mid N - Gram) \\
\quad \textit{elseif } (P(W_i \mid N - Gram) \leq ValForBad) \\
\qquad M(W_1W_2..W_m) = M(W_1W_2..W_m) \times P(W_i \mid N - Gram) \\
\textit{endfor}
\end{array}
$$

In the above algorithm, ValForGood and ValForBad are the two constants used to select those good and bad n-grams. Considering in different training corpus, the CP of good n-grams and bad n-grams may be different, we choose the values of ValForGood and ValForBad by the following steps. First, we sort descendingly all the CP estimated by the training corpus; Second, under the assumption that the first forty percent of the sorted CP corresponds to those good n-grams, the lowest CP of the first forty percent is assigned to ValForGood; Finally, we consider the last twenty percent of the sorted CP as corresponding to those bad n-grams and then the highest CP of the last twenty percent is assigned to ValForBad.

We have implemented the above two approaches to calculate fluency scores using the N-Gram model trained on the gigaWord corpus.

2. Syntactic log-odds ratio

Slor is defined as syntactic log-odds ratio (SLOR), a normalized language model score, as a metric for referenceless fluency evaluation of natural language generation output at the sentence level.

For a sentence, it can be calculated as

$$\begin{aligned}\mathrm{SLOR}(S) = &\frac{1}{|S|}(\ln(p_M(S)) \\ &- \ln(p_u(S)))\end{aligned}$$

The $P_m(S)$ can be calculated via the output of the Language model used, and $P_u(S)$ is the unigram probability of a sentence.

We have used the Gigaword Corpus, which uses the new data, with headlines as a summary of the story, a dataset for compression to train an LSTM. The embeddings have been trained by word2vec with the given description of headlines on the train dataset and have been chosen as a size of 300.

The LSTm has hidden layers of 512 neurons and 2 stacks. The output is a value of all the words that can appear next, each given a value.

The probability of a sentence has been calculated by the last fully connected layer on top of LSTM, whose values are sent through softmax and hence can be used as probabilities. Normalized by the unigram probability (since fluency should not be low if infrequent words are used) we get SLOR our first measure for fluency proposed by the paper “Sentence Level Fluency Evaluation”

Next Steps

The compression task has been always evaluated with the Rogue metric (Recall-Oriented Understudy for Gisting Evaluation) which has been used in the tasks of automatic summarization and machine translations. The Rogue metric consists of the following
Rogue-1: Based on Unigram
Rogue-2: Based on Bigram
Rogue-L: Based on the longest common subsequence

Then the next steps would be to see the correlation between the values of human-annotated values with those of the baseline metrics like rogue and SLOR and see which correlates better with the final human scores.

If time permits we would also like to try out the following:
*Use Transformers like sentence-bert (pre-trained) or trained on GigaWord(if computation is available).