Measure Text Fluency

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

Need For fluency Metrics

- Existing Metrics for evaluating models are not informative enough.
- For example in MT, for BLEU scores , "higher BLEU score was neither a necessary precondition nor a proof of improved translation Quality"
- Fully Automated metrics weakly correlate with human judgements.
- In newer tasks like unsupervised dialogue generation, where the answers are unrestricted, BLEU and ROGUE have almost no correlation as they are reference based.
- Thus a need for a new automatic metric arises, that is not reference based

Statistical Approach To Measure Fluency

- It is a statistical approach to evaluate the sentence fluency, which is based on the n-gram language model and reference-independent.
- The n-gram model is used to predict the probability of a word by the left contextual words.
- As to those word sequences not occurring in the training corpus, we use smoothing algorithm to assign them a probability

Amelioration 1

- evaluating sentence fluency using the product of n-grams' Conditional Probabilities

$$M(W_1 W_2 \ldots W_m) = [\sum_{i=1}^{m} \log(P(W_i \mid N - Gram))] / m$$

- 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

Amelioration 2

- Evaluating sentence fluency by discriminatingly treating strange and familiar n-grams

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

Amelioration 2

- In the above algorithm, ValForGood and ValForBad are the two constants used to select those good and bad n-grams.
- How to choose values for Good and Bad?
  - 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.

Results

Summary	A1_score	A2_score
prescriptions elusive for curbing microsoft	-1.015306760687785	174.3800609910745
chinese foreign minister meets with secretary-general of algerian ministry of foreign affairs	-0.6539564915788506	676.094968747561
expansion plan for ##nd street y riles neighbors	-0.7915526763150786	273.94892123422994

SLOR (Syntactic Log Odds Ratio)

- SLOR assigns to a sentence S a score which consists of its log-probability under a given LM, normalized by unigram log-probability and length:

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

- Where $P_M(S)$ is probability assigned to S under the LM, and $P_u(S)$ is unigram probability of sentence

SLOR

- The reason for normalising with unigram probabilities is that we do not want rare words to bring the score down.
- We need to divide by sentence length so that we do not prefer shorter sentences over longer ones.

(i) He is a citizen of France.

(ii) He is a citizen of Tuvalu.

Word Piece

- Since the model is too large and very difficult to train, we use a different tokenizer.
- This reduces the vocab size, reducing model size and training time.
- They also help in handling rare words since those are partitioned into more frequent segments.
- It is better than taking each character as a token where information is lost.

BaseLine Metrics

- In order to understand how good SLOR is, we need to understand how it compares to already existing metrics.
- This is done by taking the pearson correlation between the human annotated values and the values given by the metrics
- The Metrics used were ROGUE(bigram,trigram, and longest common subsequence(L)), Negative Cross Entropy and Perplexity

$$\mathrm{NCE}(S) = \frac{1}{|S|} \ln(p_M(S))$$

$$\mathrm{PPL}(S) = \exp(-\mathrm{NCE}(S))$$

Human Annotations

- Our compressed dataset contains original sentences and short paragraphs (texts) with corresponding crowd-sourced compressed versions and crowd-sourced ratings of each versions
- Each compressed versions will have corresponding ratings in terms of grammar,meaning and fluency
- The combined ratings of compressed versions will range from 6 to 24 from Most important meaning Flawless language to Little or none meaning Disfluent or incomprehensible
- These human annotated scores are correlated with different types of metrics like SLOR,WPSLOR,ROUGE-L

Results

ROGUE-L Gives a correlation 0.11 and GPT also gives correlation of 0.12, a slight increase.