Quotation Construction: A Generative Approach

Christopher Fifty (cjf92), Jingqi Zhou (jz493)

Description:

Our team's goal is to build a system that generates quotations that an individual was likely to have said, based on a corpus of text from their written works or speech. For example, our algorithm could generate a quotation for Harry Potter - one which is in line with his character - but was never actually written in the books.

Approach:

Our baseline system will consist of a simple bigram and trigram language model. Bi/trigrams are generative models which construct sentences based on word frequency. For the bigram model, the conditional probability of saying word w_i given its precreding word w_{i-1} is:

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_{i-1})}{count(w_{i-1})}$$

Recent literature associated with this problem-space implies that quotation generation can be improved by using deep learning techniques, in particular Recurrent Neural Nets. Accordingly, we intend to read the pertinent literature relating to this topic and develop a neural net with TensorFlow. We will assess the efficacy of this method and compare it with our baseline model.

Our final objective is to devise a novel algorithm which utilizes intrinsic linguistic features such as POS tags, synonyms, sentiment, etc. to augment our generated quotations. For example, it is often appropriate to substitute two adjectives (i.e. thoughtful with contemplative) in a quotation. We intend to use this technique to either augment our dataset or make direct substitutions following quotation generation with a R.N.N. or bi/trigram language model.

Evaluation:

We intend to evaluate the strength of our constructed quotations through administering surveys to classmates and friends unfamiliar with the project. A possible survey could contain machine generated quotations which are interleaved with real quotations. The participant would then be asked to identify which (if any) of the listed quotations were generated with artificial intelligence. Using this technique, we can quantify and compare the effectiveness of each quotation generation algorithm by computing a mean precision and accuracy from the survey results.

Timeline:

By April 9th (end of Cornell Spring Break):

1. Finish the bigram and trigram language models.

2. Evaluate the results of the bi/trigram language models through surveys administered to friends and classmates.
3. Finish reading primary source literature related to Neural Nets and quotation generation
4. Begin coding the R.N.N.

By May 1st:

1. Finish the R.N.N.
2. Evaluate the results of the R.N.N. through surveys administered to friends and classmates.
3. Begin devising novel algorithm for quotation generation (with focus on intrinsic linguistic features).

By May 18th:

1. Finish novel algorithm
2. Evaluate the results of the novel algorithm through surveys administered to friends and classmates.
3. Finish 10-15 page write-up.
4. Finish slide deck for presentation.

May 18th-22nd:

1. Project Presentations