Controllable text simplification

Team Name: zombies

Problem statement:

Text simplification is one of the most useful applications of NLP. Simplifying a complex text would help in many areas such as language translation, teaching kids complex texts, it helps non english speakers/ beginners understand sentences much easier. We try to build a text simplification model which has controllability where we can control the extent to which simplification happens.

Literature review:

Text simplification has been attained using various approaches from sequential models to transformers with a range of controllable parameters.
Normal seq2seq models can provide text simplification but they are highly user functionality dependent and resort to mainly deletion of words. There were few papers(Sutskever et al., 2014) which extended this model and overcame the two drawbacks.

Mounica et al., 2020 used a tool called DisSim for splitting a complex sentence and deleting the unnecessary content. Then they have used one transformer for ranking the candidates and another transformer for paraphrasing.
Louis Martin et al., 2020, used a single transformer with three major controllable tokens each for length, paraphrasing and lexical complexity.

Before these recent works, there have been attempts at simplification with only limited capabilities. Scarton and Specia (2018) and Martin et al. (2020)

added additional tokens to the input representing grade level, length, lexical, and structural complexity constraints.

There are other areas of research where they use single simplification operations.
Specia et al., 2012 research paper focused on lexical simplification which is mainly changing the complex words to simpler synonyms.
Siddharthan et al., 2006 research paper covered syntactic simplification which explicitly deals with sentence splitting.
Filippova et al., 2015 research paper focussed on sentence compression where a simpler sentences are created from complex sentences by removing irrelevant components.

Dataset:

ACL2020 folder contains the Wiki-Auto training data used in our ACL 2020 paper Neural CRF Model for Sentence Alignment in Text Simplification. train.src contains complex sentences, and train.dst contains simple sentences. This is a filtered version where we eliminate sentence pairs with high (>0.9) or low (<0.1) lexical overlap based on BLEU scores. We observed that sentence pairs with low BLEU are often inaccurate paraphrases with only shared named entities and the pairs with high BLEU are dominated by sentences merely copied without simplification. Also, this version does not contain sentence splitting. Feasibility and Metrics: The size of the dataset is very less(around 100MB). This dataset cannot be used for training a model from the scratch as the model will not encounter the diverse set of samples. Given that this dataset contains the samples without the

very high and very low lexical overlap, this is an ideal dataset for fine tuning the model after training it with another diversified dataset.

Approach:

This can be done by a normal seq2seq model right? Yes, but they are very limited with their functionalities. They are mostly used for user specific tasks. Also most of the seq2seq models restrict themselves to only deletion of the words. These cannot easily adapt to different target audiences. We try to create a hybrid model which uses splitting and deletion in sentences to give us maximum efficiency. We can also choose the degree of "simplification" we want. That is we can also choose the amount of changes we want in the output compared to the input. We will be having 3 main steps in the process of simplification. Which are candidate generation(by splitting and deleting), candidate ranking and paraphrase generation. In candidate generation, we generate a set of candidates which were based on splitting and deleting words. This can be depicted by a tree. We then give scores to each of these candidates based on their length penalised BERTscore and their compression ratio(compression ratio is the amount of words in the given candidate divided by the number of words in the original sentence). Based on the scores obtained we then take the best ranked candidate and use it for the final output. To encourage diverse outputs, we introduce a data augmentation method (It can randomly insert, delete, swap etc words in a given sentence…. link). We can have the percentage of words to be copied from input as a constraint which defines the degree of "similarity" which can be controlled. Finally we train our transformer model with the existing BERT checkpoint.

Baseline Used:

We have used the BART transformer facebook/bart-large-cnn and as our baseline models for our problem statement. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text.

BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs. The scores obtained from the baseline is:

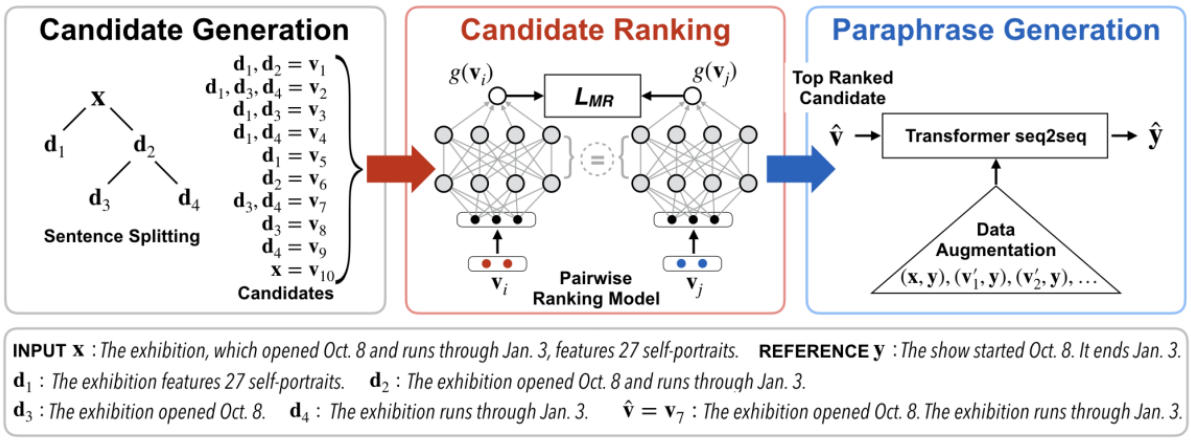
- Bert: 0.8366
- SARI: 53.33
- BLEU: 0.128

Summary of our Implementation:

We implemented this project by breaking it into 3 parts. The first part is candidate generation. In candidate generation we generate candidates which can be used as input to the transformer which generates a simplified output. We then rank the candidates that we generated based on a score which takes in the length penalty and the bertscore into consideration.

$$g^*(\mathbf{v}_i, \mathbf{y}) = e^{-\lambda||\phi_{\mathbf{v}_i} - \phi_{\mathbf{y}}||} \times BERTScore(\mathbf{v}_i, \mathbf{y})$$

We can change the length penalty as per users requirement. We then use a transformer which generates the simplified text. While training we actually give all the candidates as input for the given simplified sentence. This is called data augmentation and this helps in making the model diverse and generate different outputs. We tried copy ratio in the project by appending the actual copy ratio to the sentences, but that isn't giving good results which is potentially due to the extra token we appended in the beginning.



Quantitative Analysis:

The scores that we have obtained from the model are:

- Bert – 0.799
- SARI – 54.827
- BLEU – 0.114

Qualitative Analysis:

- Complex sentence:
  Sir George Bailey Sansom -LRB- 28 November 1883 – 8 March 1965 -RRB- was a British diplomat and historian of pre-modern Japan , particularly noted for his historical surveys and his attention to Japanese society and culture .
  Simplified sentence:
  George Bailey Sans om was 28 November 18 83 -- 8 March 1965) was a British diplomat and historian of pre - modern Japan . particularly noted for his attention to Japanese society and culture . </s>

- Complex sentence:
  The 1000s was a decade of the Julian Calendar which began on January 1 , 1000 , and ended on December 31 , 1009 .

  Simplified sentence:
  A decade of the Julian Calendar began on January 1 , 1000 . </s> .