Name: Sagar Das

Email address: sagardas7.1998@gmail.com

Contact number: 9007789474

Anydesk address: 169 667 754

Years of Work Experience: 1.5

Date: **16th October 2022**

Self Case Study -2: Deep Text Corrector

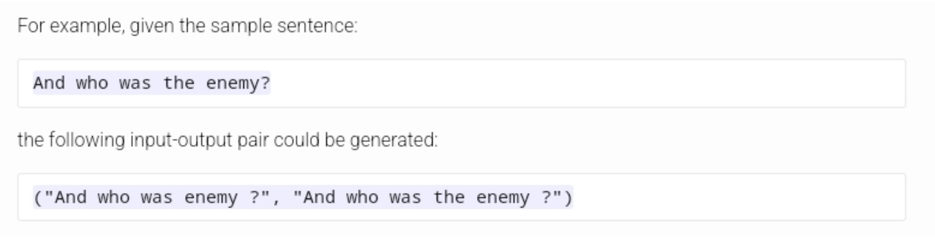
Overview

1. English is the world's most common language and is also known as the GLOBAL LANGUAGE. For developing and emerging economies, there is enormous demand and need for English in public education systems to boost stability, employability and prosperity. According to Wikipedia, English grammar is the set of structural rules of the English Language. This includes the structure of words, phrases, clauses, sentences, and whole texts.
2. While context-sensitive spell-check systems (such as AutoCorrect) are able to automatically correct a large number of input errors in instant messaging, email, and SMS messages, they are unable to correct even simple grammatical errors. For example, the message "I'm going to store" would be unaffected by typical autocorrection systems, when the user most likely intended to communicate "I'm going to the store".
3. Using the advancement in NLP using deep learning, we will try to solve this problem. For that we will be using the Cornell Movie-Dialogs Corpus, which contains over 300k lines from the movie scripts.

According to Atpatino, this is one of the largest collections of conversational written English that he could find that was almost grammatically correct.

4. **About the Corpus**:
    a. It is a large metadata-rich collection of fictional conversations extracted from raw movie scripts. In total there are 220k conversational exchanges between 10k pairs of movie characters in 617 movies.
    b. The dataset contains
        i. Speaker-level information
        ii. Utterance-level information
        iii. Conversational-level information
        iv. Corpus-level information
    c. For our project, we will not be needing all this meta-information. We will just need the Corpus level information.
5. **Creating the dataset**:
    a. In this project, we will have to create our own dataset. In order to create the dataset that is to generate input-output pairs, we will follow the next steps:
        i. We will take the sentences from Movie-Dialogs corpus.
        ii. Setting the input sequence to this sentence after randomly applying certain permutations.
        iii. Setting the output sentence to the unperturbed sentence.
    b. The perturbations applied in step (ii) are intended to introduce small grammatical errors which we would like the model to learn to correct. For this project, these perturbations have been limited to:
        i. the subtraction of articles (a, an, the)
        ii. the subtraction of the second part of a verb contraction (e.g. “‘ve”, “‘ll”, “‘s”, “‘m”)

        iii. the replacement of a few common homophones with one of their counterparts (e.g. replacing “their” with “there”, “then” with “than”)



For example, given the sample sentence:

```
And who was the enemy?
```

the following input-output pair could be generated:

```
("And who was enemy ?", "And who was the enemy ?")
```

6. So, our task for this project will be to train a neural network model on the dataset that we have created, which will be able to detect basic grammatical errors.
7. The metrics that can be used for this task are:
    a. **BLEU Score** - BLEU (BiLingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high quality reference translations
    b. **F$_1$ Score** - This emphasizes precision and recall equally.
    c. **F$_{0.5}$ Score** - This emphasizes precision twice as much as recall.

Research-Papers/Solutions/Architectures/Kernels

1. http://atpaino.com/2017/01/03/deep-text-correcter.html
    - Before training, in order to have enough data, he performed the sampling strategy multiple times in order to have a dataset 2-3x the size of the original corpus
    - He trained a neural network model consisting of LSTM encoder and decoders bridged via an attention mechanism.
    - He used a similar structure of his model as mentioned over https://arxiv.org/abs/1409.0473. Basically this paper has

implemented a RNN Encoder-Decoder Model that learns to align and translate simultaneously to translate English to French. This consists of a bidirectional RNN as an encoder and a decoder that emulates the searching through a source sentence during decoding a translation.

    - For decoding, he has added a restriction that all the tokens in the decoded sequence should either exist in the input sentence or should belong to a set of ‘corrective’ tokens. The intuition here is that the errors seen during training involve the misuse of a relatively small vocabulary of common words (e.g. “the”, “an”, “their”) and that the model should only be allowed to perform corrections in this domain.

2. Grammar as a Foreign Language:

    - The neural network model proposed by Sutsekar et al. and Bahdanau et al. to solve general sequence to sequence problems achieved state-of-art results on large scale machine translation tasks. But it was found the model by Sutsekar performed poorly on human annotated parsing datasets while model by Bahdanau performed good with a F1 score of 88.3 without use of ensemble and F1 score of 90.5 with an ensemble.

    - The attention model of Bahdanau was more data efficient and was able to get a similar score as BerkeleyParser.

    - They have used a LSTM and A Parsing Model. They have used two LSTM models, one for encoding the input sequence and another one to decode the output symbols.

- They have also used the attention model to produce each output symbol. To generate each of these symbols, it used an attention mechanism over the encoder LSTM states.
- To linearize the parsing tree, it uses a depth-first search traversal order which converts the parse tree into a linear format.

3. The CoNLL-2014 Shared Task on Grammatical Error Correction:
    - This is a paper which compares multiple submissions for the CoNLL-2014 shared task which was devoted to grammatical error correction of all error types. The error comprises articles or determiners, prepositions,noun form, verb form, subject-verb agreement, pronouns, word choice, sentence structure, punctuation, capitalization, etc. In total there are 28 types of errors that have been taken into account over here.
    - In total there were 13 teams that made their submission. Most of the teams built a hybrid system that made use of a combination of different approaches to identify and correct errors.
    - Most of the teams followed this basic approach:
        i. the probability of a learner n-gram is compared with the probability of a candidate corrected ngram, and if the difference is greater than some threshold, an error was perceived to have been detected and a higher scoring replacement n-gram could be suggested

        ii. Some teams used this approach to detect errors only and then corrected the errors using some other methods.

        iii. While the other teams used different methods to detect the errors first and then made corrections based on the alternative highest n-gram probability scores.

- To correct the errors the following approaches were mainly used:

    i. To use a phrase-based statistical machine translation system to translate learner English into correct English.

    ii. Also in regard to correcting single error types, rules-based approaches were also common in most teams' approach because some error types were more common than others and so in order to boost the accuracy simple rules can be written for that.

    iii. Some teams also used various classifiers to correct specific error types.

- **$F_{0.5}$** was used as the evaluation metric in the CoNLL-2014 shared task instead of F_1 used in CoNLL-2013. F_{05} emphasizes precision twice as much as recall, while F_1 weighs precision and recall equally.

4. Constituency Parsing with a Self-Attentive Encoder:

    - This paper demonstrates that replacing an LSTM encoder with a self-attentive architecture leads to improvements to a state-of-the-art discriminative constituency parser. They have introduced a parser that combines an encoder built using self-attentive architecture with a decoder customized for parsing.

    - Their model learns to use a combination of the two types of attention types, with position-based attention being most important. Content based attention is more useful in the later layers.

- Using windowed attention, that is strict windowing yields poor results.

First Cut Approach

Based on the research and readings that I have done, I will follow the following steps -

1. Using the Movie Corpus dataset, we will have to create our own dataset by extracting the corpus level information and applying certain permutations. The steps to make the permutations have been mentioned earlier. We will have to repeat the steps multiple times to get a large enough dataset.

2. Once the dataset is created, we have to build our model. And to build our model, we will combine the concepts from Paper I and Paper II. That is we will use a RNN based Encoder Decoder Model with an attention mechanism. Also from paper II, we will try to implement a parsing model along with LSTM as that improved the score quite significantly in their case.

3. So, we will try to implement LSTM and A Parsing Model. We will use two LSTM models, one for encoding the input sequence and the other one to decode the output symbols. We will also use the attention mechanism to generate the output symbols over the encoder LSTM states.

4. As we are using A Parsing Tree Model, we will need to linearize the graph-like structure and for that we will follow the depth-first search traversal order.

5. For scoring, we use $F_{0.5}$ as mentioned in Paper III. And for loss function we can use categorical cross-entropy.