Project

Character-level Recurrent Text Prediction

Zehui Xuan, Pierre Onghena, Yoann Lemesle

Contents

1 Introduction

There is a huge growth of AI systems and their impact, not only with respect to the academic field, but also with respect to the evolution of societies and industries. The author M. Low exploits the possibility of text prediction to mobile phones [2]. As these devices are constrained with limited computational power, only simple statistical models are considered feasible to perform the text prediction task. Therefore, the paper seeks to avoid the implementation of high dimensional embedding matrices by shifting to character-level architectures. However, as the latter gives up the semantic information of words, it entails a smaller vocabulary with the promise of capturing the underlying sentence structure.

2 Methodology

2.1 Research Questions

The project wants to provide research on different approaches to target the limitations of character-level text prediction. By conducting experiments, the project attempts to address and answer the following research questions:

1. Is it possible to mimic the performance of the initial LSTM network by implementing the structure?

2. Is it possible to apply a transformer model on the character-level text prediction?

2.2 Dataset

The Brown corpus is designated within this setting of research as it provides a baseline for conducting experiments against different methods [1]. Moreover, the dataset provides a quantitative evaluation to benchmark changes in the network architecture and assess the performance. In this project, we divide the corpus into train / valid / test sets in the proportion of 20% / 10% / 70%.

2.3 Methods

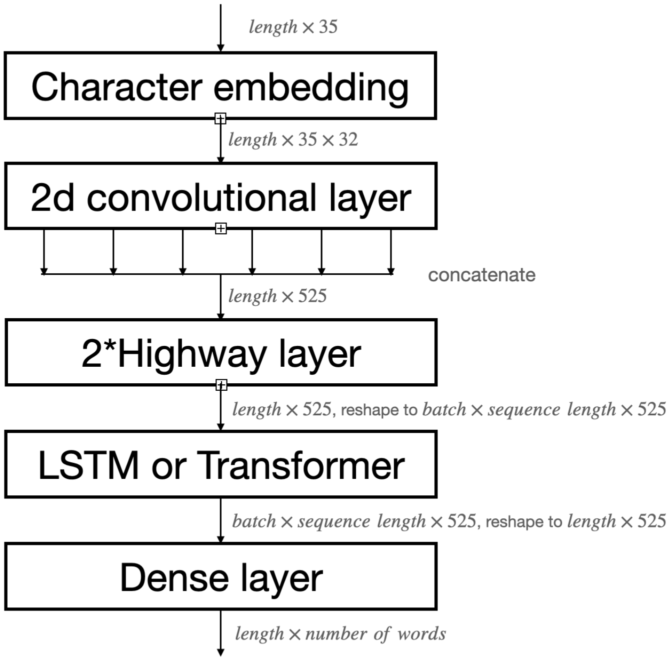

Figure 1: Structure of models (batch norm and dropout layers omitted)

2.3.1 Data Preparation

A pipeline is set up to split the corpus into batches for more efficient training. The text needs to be vectorized in order to reflect the linguistic properties as inputs for the model. But instead of using words to build a word embedding, the project goes one level deeper at the character level. Here we vectorized each word in the corpus, based on the letters or symbols they contained. Thus each word is transformed into a vector of length 35. (The longest word contains 33 letters and the empty place in other shorter words are filled by padding marker. Each word also is also added a beginning marker and an ending marker.)

2.3.2 Character-level embedding layer and CNN

In this project, the model begins with a Character-level embedding layer, which can look up the embedding of size 32 for each character. Then each word after this step is transformed into a 2d tensor of shape (35,32).

Although word level models have proven to capture information well in the large embedding space, character level requires a smaller amount of embedding (60 in Brown Corpus, for all the letters, punctuation, and markers). However, the character level model has to deal with larger sequences in order to form a word. Therefore, it entails a space-time tradeoff as a character model captivates less memory but involves more computational power to construct a word sequence.

A 2d convolutional layer follows the embedding layer, composed by kernels of different sizes. The results are concatenated then form a vector of length 525 for each word. Then the tensor of the sequence enters either the LSTM model or the transformer model, both with highway layers, which will be described in next sections.

2.3.3 LSTM

An LSTM model can be applied to a variety of time-series input data. For example, it can be used to to construct an autoencoder for anomaly detection. Moreover, a text represent a sequence of data where the order matters. Therefore, an LSTM can give good results as the long and short-term memory can learn the relationship of variables to recognize patterns of linguistic properties.

The inspired paper proposed an LSTM architecture with stacked layers of size 512, batch normalization and dropout as regularization techniques. Furthermore, it mentioned that adding a highway layer could improve the model further. Therefore, this project applied the general architecture setting with two highway layers before the LSTM layers to reproduce results.

2.3.4 Transformer

As stated in the risk analysis of the project plan, it's necessary to identify the feasibility of implementing different methods. More precisely, as the research of the initial paper is based on LSTM, it's unclear if and how the recently more popular but also more complex transformer approach would perform on the level of character text prediction.

Inspiration for the transformer network with attention mechanism was found within one of the PyTorch tutorials[1] and guide[2].

3 Evaluation

The cross-entropy loss is applied to minimize the KL-divergence of the learned and empirical character distribution. Over 20 epochs during the training process, it can be seen from the left side of figure 2 and 3 that the learning entails a good fit. In specific for both models, the training and validation loss decrease with a small generalization gap.

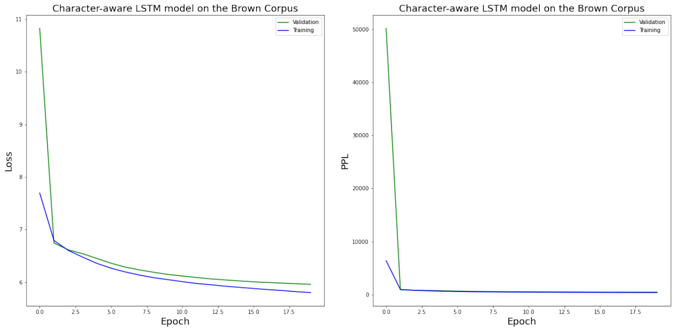

Figure 2: Character LSTM model

In addition, the metric of perplexity (PPL) is used to assess the performance of the network. However, as stated in a blog post[3], the optimal lower bound value of perplexity is unknown. Moreover, it also mentions that a character-level model with smaller perplexity than that of a word-level doesn't necessarily induce a better model. Although the comparison between character and word architectures isn't the research objective, this project seeks the implementation and evaluation of an LSTM and transformer model at character-level.

[1] `https://pytorch.org/tutorials/beginner/transformer_tutorial.html`

[2] `https://towardsdatascience.com/a-detailed-guide-to-pytorchs-nn-tran`

[3] `https://thegradient.pub/understanding-evaluation-metrics-for-langua`

model	train	valid	test
in original article	147	291	320
ours (LSTM)	350	426	954
ours (Transformer)	514	554	1155

Table 1: Perplexity comparison between models on Brown corpus

From the right side of figure 2 and 3, it's illustrated that the perplexity rapidly drops to lower levels. However, the LSTM model reaches a smaller perplexity value of 350 at the last epoch than the transformer model with a value of 514.

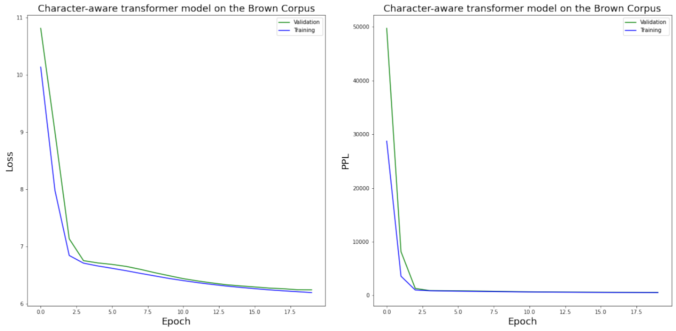

Figure 3: Character Transformer model

4 Conclusion

In this project, we mimic a character-level LSTM network for text prediction. We further replace the LSTM part with Transformer. The result shows that Transformer can be used in the character-level prediction model, but it currently does not perform as well as the LSTM model, so it still has improvement space. Also, our imitation is not able to surpass the model of the original article, which may be due to the limitation in computational power.

References

[1] W. N. Francis and H. Kucera. *Brown Corpus Manual*. Tech. rep. Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979. URL: `http://icame.uib.no/brown/bcm.html`.

[2] Melvin Low. "Character-level Recurrent Text Prediction". In: 2016.