Punctuation Prediction for Audio Speech Transcripts

Sumedha Raman
Georgia Institute of Technology
Atlanta, GA, USA
sraman46@gatech.edu

Mohit Srinivasan
Georgia Institute of Technology
Atlanta, GA
mohit.srinivasan@gatech.edu

Cesar Santoyo
Georgia Institute of Technology
Atlanta, GA
csantoyo@gatech.edu

Abstract

Transcripts generated from audio or video files require proper punctuation to be readable. In this paper, we investigate the use of different deep learning architectures for automated punctuation of transcripts. Specifically, we construct five different architectures which are trained on punctuated transcripts sourced from TED talks. Each architecture is explored in detail; we highlight the benefits and deficiencies of each one by analyzing each architecture's performance in punctuation prediction with three chosen punctuation marks. Furthermore, we analyze the effects of the data utilized on the performance of the neural network architectures.

1. Introduction

Proper punctuation is useful for improving accessibility to information in various domains that deal with audio and video data. Specifically, online education, entertainment and media, law, can become more accessible to people who heavily rely on transcriptions. While these transcripts are often written manually, they can be generated in an automated fashion as well. Transcripts of speeches derived from pure audio or video format can be generated using Automatic Speech Recognition (ASR). While ASR is effective at generating text transcripts of audio, it often does not punctuate the output text. This can lead to less than optimal transcriptions of audio, e.g, podcasts, TED talks, seminars, etc., that can be difficult to parse by a human reader. Transcriptions are of particular importance to people with hearing disabilities, if they cannot make use of the original audio. In addition, poor punctuation hinders the ability to use the derived transcript as a useful dataset for training networks for tasks such as language translation.

Previous work addresses the problem of punctuation restoration by using long short-term memory (LSTM) [15] or bi-redirection recurrent neural networks (RNNs) [16]. In [15] the authors focus on solely implementing a recurrent neural network to apply periods and commas to transcripts which are not punctuated; [15] derives their LSTM framework from [13]. Here, the authors use a two stage model where the authors combine textual and speech-pause information as stages which inform the neural network where to punctuate. In [16], the authors approach addresses the punctuation problem by leveraging bi-directional LSTMs and implementing an attention mechanism which increases the network's ability to find relevant context for punctuation decisions.

Furthermore the paper [9] uses OpenNMT [10] with a bi-directional RNN architecture while simultaneously utilizing parts of speech (POS) tags to make punctuation predictions. Specifically, in [9], the bi-directional RNN is the first layer which is implemented as an LSTM while the POS tags are a second layer which adds syntactic information. In [18], deep bi-directional LSTMs are studied to understand whether adding layers significantly improves the performance of a neural network.

In this paper, we implement a series of NN architectures to achieve this task. Specifically, we implement a bi-directional LSTMs, Gated Recurrent Unit, Transfer Learning, and variations coupling each of these techniques. We aim to improve the performance of past implementations while also implementing punctuation beyond commas and periods. This paper is organized at follows: Section 2 covers the methodology, particularly the architectures of the neural networks of interest, Section 3 contains the results of the various neural network implementations as well as a detailed discussion of those results. The discussions elaborate on observations, and potential future improvements that

can be made to the code base. Lastly, Section 4 concludes the paper.

2. Approach and Methodology

In this section, we describe the datasets we are using as well as the five different implemented neural network architectures. A brief structural characterization of each architecture is provided. We implement the networks using Pytorch [12] on Google's Colaboratory.

2.1. Datasets

For this paper, we utilize the IWSLT 2012 dataset [2] which consists of punctuated transcriptions of TED talks. This dataset is utilized in the IWSLT Evaluation Campaign for three research tasks: automatic transcription, speech translation and text translation. Additionally, it is used for the punctuation prediction task in [16] and [3], where it is described in more detail in the latter. The IWSLT Machine Translation (MT) track consists of training, development and evaluation data which total 2.4M words, all of which are used in this work.

To preprocess this dataset, we choose to remove certain punctuation symbols and limit the data to only contain punctuation we are interested in, i.e., commas, periods and question marks. Moreover, the data is split into fixed-size segments during preprocessing. Specifically, the sentences were saved in segments that were of a particular length. In case a sentence is incomplete in a certain segment, the entire sentence is added to the next segment. However, sentences which are longer than the segment size are discarded. Each word is considered to be followed by a punctuation tag, with the possible tags being comma, period, question mark and none. We implement a random 70-20-10 split of the preprocessed data segments to generate the training, validation, and test sets. When using a segment size of 32, this results in a training, validation, and test dataset of length 49k, 14k and 7k, respectively.

We choose to utilize transfer learning as one of the network architectures which requires the usage of an additional dataset from a different domain. Hence, for transfer learning, the data is comprised of New York Times articles sourced from the New York Times archive [1]. The New York Times archive API is used to access articles in the archive; subsequently, we preprocess the article text in the same manner as described above. A dataset of 128k instances (each with segment size 32) is used to pre-train the transfer learning model.

2.2. Neural Network Architecture

We propose to evaluate the effectiveness of different LSTM architectures for the punctuation prediction task. Here, we choose to implement architectures using LSTMs, BLSTMs (bidirectional LSTM) and GRUs. Additionally, we also implement transfer learning by pretraining on a large dataset and fine-tuning on the training set. The model architectures implemented are described below. The Adam optimizer was used for all architectures, and the loss function used was the negative log likelihood (NLL Loss).

2.2.1 LSTM

Long Short Term Memory (LSTM) Networks are a type of RNN (Recurrent Neural Network) that can store long-term temporal dependencies. They are well suited for classification, prediction, and processing when dealing with time-series data. LSTMs have been applied to a variety of areas such as robot control [8], time series prediction [6], speech recognition [14], sign language translation [7], and punctuation [15].

Recently, LSTMs have been used for punctuation prediction [15] with a reasonable degree of accuracy. This type of network is well-suited for punctuation prediction, since the punctuation symbol after a certain word can depend on a variable number of prior words.

For the task of automatic punctuation, the architecture of the network we use consists of an embedding layer, an LSTM layer and a linear layer, followed by an application of the softmax function. The dimensions of the embedding layer and hidden layer are 1024 and 256, respectively.

2.2.2 BLSTM

The bidirectional LSTM architecture consists of two LSTMs — one processes the input sequence as is, and the other processes the reverse of the input sequence. This is important for punctuation prediction because the symbol after a word can depend on the words following it in addition to the words preceding it. An illustration of punctuation prediction using a BLSTM is depicted in Figure 1. In Figure 1 we see that the initial sentence passes through the BLSTM and ultimately the network classifies whether each individual word is followed by a particular punctuation mark or none at all. Based on the results obtained in [18], we also tested an architecture consisting of two BLSTM layers. This particular case is denoted as 2 BLSTM in Table 1.

2.2.3 Gated Recurrent Unit

Gated Recurrent Unit (GRU) is an RNN architecture that is similar to Long Short Term Memory (LSTM) networks which was first introduced in [4]. A general GRU is illustrated in 2. GRUs are simpler than LSTM networks, and have been shown to work reasonably well with certain small datasets [5]. However, GRUs try to tackle the vanishing gradient problem by employing an update gate and a reset gate. GRUs have been used for punctuation of transctipts [5], and

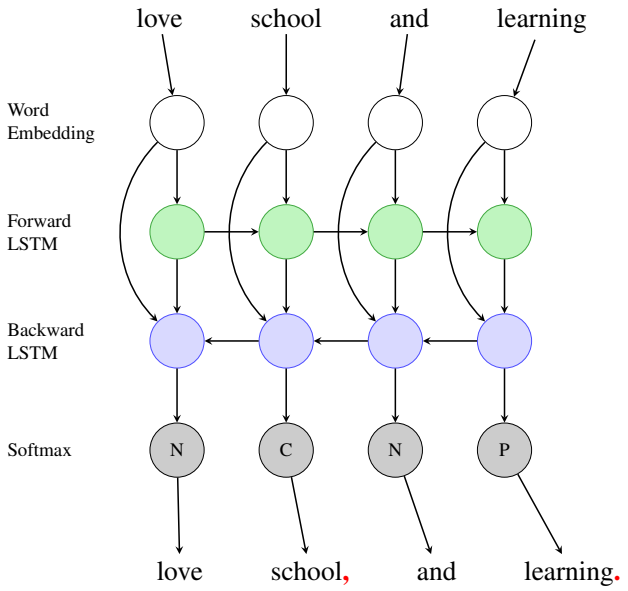

Figure 1. Bi-directional LSTM Illustration of punctuation classification. Here, we see the the network classifies whether there should be a period, comma, question mark, or no punctuation after a particular word in the sentence. Specifically, this examples illustrates a comma being added after the word "school" and a period after the word "learning".

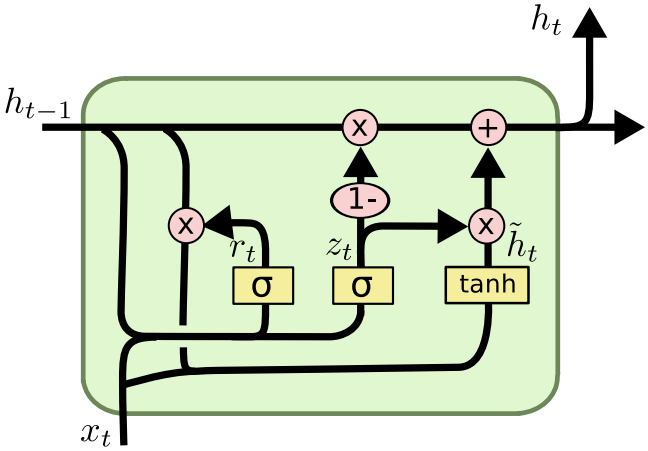

Figure 2. An illustration of a Gated Recurrent Unit (GRU) from [17]. A key feature of the GRU is it supports the gating of the hidden state. While related to LSTMs, it is a much simpler architecture than LSTMs.

also have been used in conjunction with other architectures for punctuation [16]. In this paper, we use a stand alone GRU architecture consisting of 8 layers along with a linear layer and show that the network performs reasonably well for most symbols. The results are detailed in Table 1.

Architecture	Period	Comma	Question	Overall
LSTM	0.324	0.434	0.084	0.334
BLSTM	0.569	0.574	0.182	0.527
2 BLSTM	0.594	0.560	0.154	0.520
GRU	0.565	0.583	0.183	0.527
BLSTM-Transfer	0.606	0.560	0.161	0.526

Table 1. F1 Scores for the various architectures tested

2.2.4 Transfer Learning

The availability of punctuated speech transcripts is limited, but there is no dearth of punctuated text available online. We hypothesize that using a larger dataset, which is not necessarily speech-based text, for pretraining the network followed by fine-tuning on speech-based text could improve the results obtained in punctuating speech-based text.

Hence, we choose to utilize transfer learning as one of the architectures through which we implement the automatic punctuation mechanism. Transfer learning contrasts traditional supervised learning in that it allows for different domains to be utilized during the training and testing of the network [11].

Specifically, the network architecture we use for transfer learning is the same as that of the BLSTM model.The model is initially trained on the NYT data, and subsequently fine-tuned on the TED transcripts data. The results obtained using transfer learning are listed as BLSTM-Transfer in Table 1.

3. Results and Discussion

The neural network architectures presented in Section 2.2 were implemented and trained on the datasets described in Section 2.1. The models were evaluated on the precision, recall, and F1 scores; this was calculated per punctuation and overall. This metric was chosen because of the class imbalance — the number of words with no punctuation far outnumbered those with any of the three symbols. The F1 scores obtained for each of the architectures is shown in Table 1. For reference, the state-of-the-art model that predicts the same set of punctuation symbols produced an overall F1 score of 0.63 for this task [16].

As seen in Table 1, the results obtained using the LSTM model was subpar compared to the five other architectures tested. The other architectures (all bidirectional) produced similar results, with overall F1 scores of 0.52. Unlike [18], we did not see a significant improvement when two BLSTM layers were used instead of one, although the F1 score for the period symbol was higher. The GRU architecture achieved very similar results in comparison with the BLSTM architecture. With transfer learning, the F1 score of the period symbol increased, which could indicate that the punctuation style for written and spoken word is different for commas and question marks, but similar for periods.

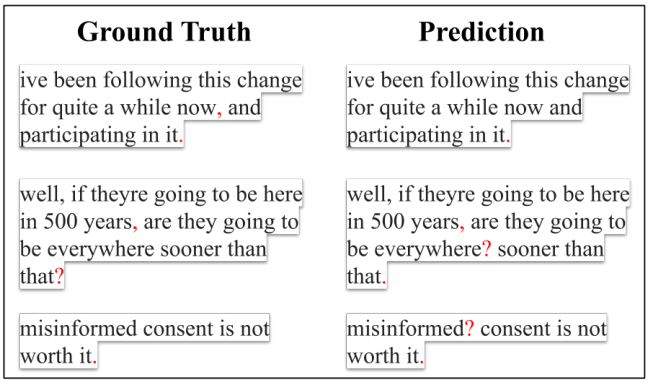

Figure 3. Examples of punctuation mismatches.

Hyperparameter	Value
Embedding Dimension	1024
Hidden Dimension	256
Learning Rate	10^{-4}
Batch Size	16
Segment Size	32

Table 2. Model hyperparameters which provide the best model performance across the network architectures of interest.

Dataset	Period	Comma	Question Mark
TED Transcripts	133,970	165,002	11,388
NYT Archive	241,659	29,1813	3,301

Table 3. Punctuation Symbol Counts in Datasets.

It was found that all of the models performed poorly on question mark prediction. In particular, the precision values of the models for the question mark symbol were very low, which shows that there were a large number of false positives. One hypothesis for this poor performance is the class imbalance. An analysis of the dataset showed that the number of question mark symbols in the text was far fewer than that of periods and comma, as shown in Table 3. For the TED transcripts dataset, the number of periods and commas were more than 10 times the number of question marks. This ratio is even more skewed for the NYT dataset. The BLSTM-Transfer model produced a lower F1 score for question marks than the other BLSTM models, which supports this theory. A large dataset with a similar ratio of punctuation symbols may be more useful for transfer learning. The authors of [16] noted that using attention mechanism helped improve the F1 score of question marks, because focusing on certain words such as "what", "how", "where", and so on can be good indicators of questions.

Figure 3 demonstrates some examples of predictions made using the BLSTM model and the corresponding ground truth punctuation. In the first example, the predicted punctuation does not match the ground truth precisely, but the prediction is well formed and the sentence coveys the intended meaning. This output would be acceptable since the main focus is to improve readability of the text without modifying the meaning. The prediction in the second example is an acceptable style of speech-based text, but modifies the intended meaning to a certain degree. The third example is also acceptable in form, but the meaning is altered significantly. This instance exemplifies the difficulty of the task — predicting valid punctuation is different from predicting punctuation that fits the context.

Another interesting finding during the experiments was that a relatively small segment length of 32 produced better results than larger segment lengths. Our initial presumption was that a larger segment length would be better, since it provides more context while determining the punctuation symbol that follows a word. Additionally, speech text can have long sentences, which would be ignored while training when a small segment length is used. However, we found that a segment length of 100 resulted in an F1 score of 0.49, where as a segment length of 32 for the same architecture resulted in an F1 score of 0.52. One plausible explanation for this could be that with smaller segments, there are fewer sentences per segment, making it easier for the model to learn phrase and sentence boundaries.

4. Conclusion and Future Work

In this paper, we study the problem of punctuation prediction for speech text by using various RNN architectures. Each RNN architecture is chosen based on an extensive literature review; however, some of the architecture choices were taken to approach the problem of automatic speech translation from a perspective not previously published. The results obtained using the various architectures were analyzed and compared using standard performance metrics. Generally, our results don't outperform the state of the art; however, we identify plausible reasons behind this performance shortcoming related to the network architectures and the chosen datasets.

Based on our results and prior work, some improvements can be made to boost the performance of models on the task. Attention mechanism can be explored for improving the results while predicting question marks. Using audio data, specifically pause time between words, could be one way to approach the problem of predicting valid punctuation that alters the intended meaning. Another area to explore would be using intonation in speech audio audio for punctuation prediction. Additionally, if the model could learn the context of the speech, major changes in meaning could be avoided. Other potential areas of future work include using a more balanced dataset, and adding additional architectures such as attention mechanisms to focus on more sensitive punctuation such as question marks.

References

[1] The new york times web archive. https://archive.nytimes.com/. 2

[2] M. Cettolo, C. Girardi, and M. Federico. Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation*, pages 261–268, 2012. 2

[3] X. Che, C. Wang, H. Yang, and C. Meinel. Punctuation prediction for unsegmented transcript based on word vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 654–658, 2016. 2

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[6] F. A. Gers, D. Eck, and J. Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002. 2

[7] D. Guo, W. Zhou, H. Li, and M. Wang. Hierarchical lstm for sign language translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[8] D. N. T. How, K. S. M. Sahari, H. Yuhuang, and L. C. Kiong. Multiple sequence behavior recognition on humanoid robot using long short-term memory (lstm). In *2014 IEEE international symposium on robotics and manufacturing automation (ROMA)*, pages 109–114. IEEE, 2014. 2

[9] C. C. Juin, R. X. J. Wei, L. F. D'Haro, and R. E. Banchs. Punctuation prediction using a bidirectional recurrent neural network with part-of-speech tagging. In *TENCON 2017-2017 IEEE Region 10 Conference*, pages 1806–1811. IEEE, 2017. 1

[10] G. Klein, Y. Kim, Y. Deng, V. Nguyen, J. Senellart, and A. Rush. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184, Boston, MA, Mar. 2018. Association for Machine Translation in the Americas. 1

[11] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 3

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[13] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014. 1

[14] H. Soltau, H. Liao, and H. Sak. Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975*, 2016. 2

[15] O. Tilk and T. Alumäe. LSTM for punctuation restoration in speech transcripts. In *Interspeech 2015*, Dresden, Germany, 2015. 1, 2

[16] O. Tilk and T. Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*, 2016. 1, 2, 3, 4

[17] F. Visin. Deep recurrent neural networks for visual scene understanding, 2016. 3

[18] K. Xu, L. Xie, and K. Yao. Investigating lstm for punctuation prediction. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2016. 1, 2, 3

Work Division

The delegation of work among team members for this project is described in Table 4.

Student Name	Contributed Aspects	Details
Sumedha Raman	Data Preprocessing, Implementation and Analysis	Preprocessed data into segments, implemented Transfer Learning, analyzed results of models, worked on report
Mohit Srinivasan	Implementation and Analysis	Implemented the GRU, BLSTM-GRU architectures, literature review, Hyperparameter tuning, worked on report
Cesar Santoyo	Implementation and Analysis	BLSTM/Two BLSTM and analyzed the results, implemented randomized train, val and test sets, hyperparameter tuning, report

Table 4. Contributions of team members.