HIGH SCHOOL EDITION
Journal of Student Research

Punctuation Restoration for Speech Transcripts using seq2seq Transformers

Aviv Melamud[1] and Alina Duran[#]

[1]Cresskill High School, Cresskill, NJ, USA
[#]Mentor

ABSTRACT

When creating text transcripts from spoken audio, Automatic Speech Recognition (ASR) systems need to infer appropriate punctuation in order to make the transcription more readable. This task, known as punctuation restoration, is challenging since punctuation is not explicitly stated in speech. Most recent works framed punctuation restoration as a classification task and used pre-trained encoder-based transformers, such as BERT, to perform it. In this work, we present an alternative approach, framing punctuation restoration as a sequence-to-sequence task and using T5, a pretrained encoder-decoder transformer model, as the basis of our implementation. Training our model on IWSLT 2012, a common punctuation restoration benchmark, we find its performance is comparable to state of the art classification-based systems with an F1 score of 80.7 on the test set. Furthermore, we argue that our approach might be more flexible in its ability to adapt to more complex types of outputs, such as predicting more than one punctuation mark in a row.

Introduction

Automatic speech recognition (ASR) is used in various applications, such as the preparation of medical reports, hands-free typing of documents, implementation of voice-based user interfaces and virtual assistants, automatic transcription of videos/lectures, and accessibility tools. The field of ASR has seen massive progress in recent years. State of the art (SOTA) models, such as Wav2Vec 2.0 (Baevski et al. 2020), have achieved extremely low word error rates (WER) on ASR benchmarks such as “TIMIT” (Garofolo et al. 1993) and “LibriSpeech” (Panayotov et al. 2015), measured at 8% and 1.4% Word Error Rates (WER), respectively. WER counts the percentage of erroneous substitutions, deletions, and insertions of words in an ASR-generated transcript. While word accuracy is important for a quality transcript, missing punctuation, not evaluated in WER metrics, has been shown to impact readability just as much as word errors (Tündik et al. 2018). Due to the fact that punctuation is less explicitly indicated in speech, it is typically inferred from context and has been addressed using different approaches. This task is known as punctuation restoration.

Previous research on punctuation restoration has utilized Recurrent Neural Network (RNN) architectures (Tilk and Alumäe 2016), and LSTMs (Tilk and Alumäe 2015), while most recent attempts have focused on using transformer models (Nagy et al. 2021), due to their superior performance. Specifically, pretrained contextualized language models, such as BERT (Devlin et al. 2018), fine-tuned on the punctuation restoration task, have yielded state of the art performance (Courtland et al., 2020).

Common to most of the recent work is the framing of punctuation restoration as a classification task, where a single punctuation mark is predicted for every position in the input text. In contrast, in this paper, we frame the task as a textual sequence-to-sequence task applying pre-trained seq2seq transformer models, specifically Google’s T5 (Raffel et al. 2020), to the punctuation restoration of speech transcripts.

Related work

Early punctuation restoration systems relied on classical machine learning methods, such as decision trees (Kolár et al. 2004) and n-gram models (Gravano et al. 2009). However, these have largely been outperformed by large neural network models. Since the task requires a model well suited to processing sequences, with an understanding of context, Recurrent Neural Network (RNN) architectures have been used to restore punctuation, including RNNs with attention (Tilk and Alumäe 2016), and LSTMs (Tilk and Alumäe 2015). Most recent literature focuses on transformers, due to their superior contextual understanding and accuracy as compared to LSTMs and RNNs. Transformer encoders pre-trained on large corpora of text, such as BERT (Devlin et al. 2018) and RoBerta (Liu et al. 2019), have obtained state-of-the-art results in a variety of natural language tasks, and have thus been applied to punctuation restoration (Nagy et al. 2021, Alam et al. 2020).

Common to most of the aforementioned models is the framing of punctuation restoration as a classification task, where for every position in the input text, the model makes a classification decision between one of several predefined classes (period, comma, question mark, none, etc.). Figure 1 illustrates this approach, where every position in the input is encoded by the encoder and then outputs are predicted based on that encoding.

Previous encoder-only architectures (Nagy et al. 2021) pass this vector to a classifier, which classifies each token of the input as being followed by one of several punctuation types as seen in figure 1.

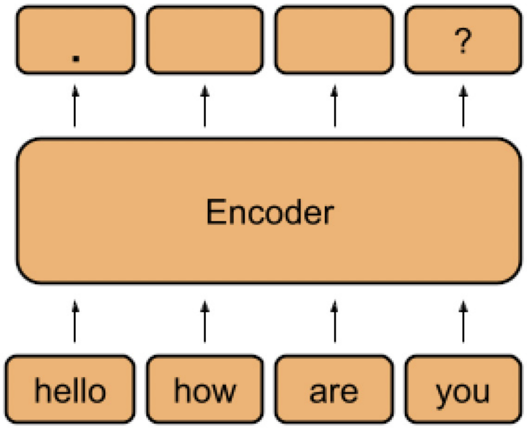

Figure 1. Illustration of encoder-classifier model for punctuation restoration.

In this paper, we extend this line of research, by framing punctuation restoration as a sequence-to-sequence task rather than classification. To do this, instead of using a single encoder, we train an encoder-decoder sequence-to-sequence transformer model. Our goal is to determine the efficacy of using an encoder-decoder architecture instead of an encoder-only system. Figure 2 illustrates the sequence-to-sequence encoder-decoder architecture used in this paper.

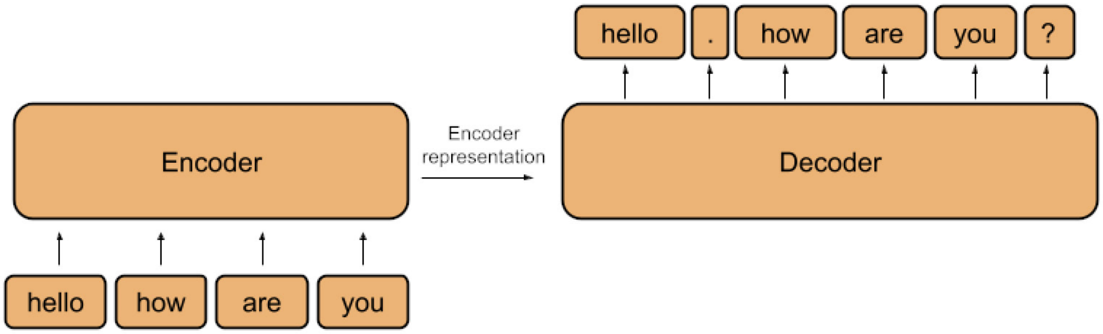

Figure 2. Illustration of encoder-decoder seq2seq model for punctuation restoration.

We note that a couple of previous works used machine translation approaches to address punctuation restoration (Peltz et al. 2011, Vāravs and Salimbajevs 2018). Similar to this work, these past works frame the punctuation restoration task as a text-to-text task. However, our work is based on a simple vanilla sequence-to-sequence architecture and newer pre-trained models that are more comparable to the models used by the recent state-of-the-art work.

Methods

Model

We chose to use Google's T5 (Raffel et al. 2020), a sequence-to-sequence encoder-decoder model, to perform punctuation restoration. T5 uses a bi-directional transformer, similar to BERT, as its encoder, and an autoregressive transformer decoder. It was trained to perform seq2seq tasks on 20T of the "Colossal Clean Crawled Corpus" (Raffel et al. 2020).

As illustrated in Figure 2, when fed a sequence of tokens (unpunctuated text), T5 passes the input through an encoder which generates a vector representation of the sequence. Then it passes that vector representation to a decoder, which generates an output sequence of tokens. This architecture was shown to be a good fit for various tasks from machine translation, to text summarization and question answering. In our case, to perform punctuation restoration, the input sequence is the speech transcription without punctuation, and the output sequence is a fully punctuated version of that input. Unlike the case of the encoder-classifier model, described in Section 2, the seq2seq architecture does not impose any explicit constraints on the relation between the input and the output structures. Specifically, the output can be of arbitrary length, seamlessly allowing the generation of two or more punctuation marks one after the other (as in "That is incredible!!!").

While T5 was originally implemented in the Mesh TensorFlow library (Shazeer et al. 2018), we utilize the PyTorch implementation provided in Huggingface's transformers library (Wolf et al. 2019).

Methodology

We finetune the pre-trained T5 model on a dataset of Ted talk transcripts (Federico et al., 2012). The problem is formulated as a text-to-text task, where the input is an uncased segment of a Ted transcript devoid of punctuation, and the ground truth output is the original punctuated text (also uncased), as illustrated in Table 1. The text was broken into 256-token long sequences. In order to most closely resemble the data T5 was pre-trained on (leveraging the information it learned during training), we frame the task as sequence to sequence, keeping the original tokens used for each punctuation character.

Table 1. A shortened example of the input-output pairs fed into the model during training.

Input	we know that right we've experienced that
Ground truth output	we know that, right? we've experienced that.

Experimental Details

Datasets

We trained on the IWSLT 2012 English Ted Talk dataset (Federico et al., 2012). This dataset is commonly used as a benchmark for punctuation restoration models. It comprises 1,066 transcripts of Ted talks, and 2.4M words in total and is split into a 2.1M word training set (87.5%), a 296K validation set (12.3%), and a 12K word test set (0.5%). We used this original split in our experiments. Table 2 shows the distribution of labels in this dataset.

Table 2. Distributions of labels of in IWSLT 2012 the dataset.

	Train	Validation	Test
Period (.)	139,619	909	1,100
Comma (,)	188,165	1,225	1,210
Question mark (?)	10,215	71	46
None (Word not followed by punctuation)	2,001,462	15,141	16,208

Training

We perform gradient descent using the Adafactor optimizer (Shazeer and Stern 2018) used for fine tuning in the T5 paper (Raffel et al. 2020), with a learning rate of 3E-4 and weight decay of 0.1 utilizing cross-entropy as the loss function. Of the 5 variants of T5 released by Google, we only fine tune the 3 smallest as seen in Table 3. For the T5-small and T5-base model we use a batch size of 16, while a batch size of 12 is used for T5-large, due to limitations in GPU memory (VRAM). Gradient checkpointing (Rajbhandari et al., 2019) was also used to reduce VRAM requirements on large model training. Training continued until a minimal loss was achieved on the validation set. All experiments were conducted on an NVIDIA Tesla P100 GPU.

Table 3. Parameter count for T5 variants used in this paper

T5 variant	Parameter count (approximate)
T5-small	60M
T5-base	220M

T5-large	770M

Evaluation

The test set was split into 50 batches and fed into the model. Following previous work (Courtland et al., 2020, Nagy et al. 2021), the output was evaluated with an F1-score over 3 classes (comma, period, question mark). As a baseline, we show the results of Courtland et al., 2020, a BERT-based encoder model, which is considered state of the art for this dataset. The model they use has 110 million parameters, which is in between our T5-small and T5-base.

The results are reported in Table 4, showing our T5-base and T5-large models with an overall performance that is comparable or slightly better than the BERT baseline.

Results

Table 4. Precision, recall, and F1-score on the IWSLT 2012 Ted dataset

	Period (.)			Comma (,)			Question mark (?)			Overall		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1s
BERT-base (Courtland et al., 2020)	72.8	70.8	71.8	81.9	86.4	80.8	80.8	91.3	85.7	78.5	82.9	80.6
BERT-base-uncased (Nagy et al. 2021)	59	80.2	68	83	83.6	83.3	87.8	83.7	85.7	76.6	82.5	79
Albert-base (Nagy et al. 2021)	55.3	74.8	63.6	76.8	87.9	82	70.6	83.7	76.6	67.6	82.1	74.1
T5-small	77.9	83.2	80.5	72.9	55.5	63	70.7	74.4	72.5	75.6	69.3	72.4
T5-base	84.8	89.2	86.9	78.3	70.8	74.4	67.3	84.6	75	81.3	80.1	80.7
T5-large	86.8	90	88.4	77.3	73.2	75.2	78.3	92.3	84.7	82.1	81.8	82

Figure 3 shows a confusion matrix between the different classes ('NONE' means predicting that no punctuation mark is required). Interestingly, it shows that one of the common mistakes that the model makes is with the placement of commas, which is typically challenging for humans as well.

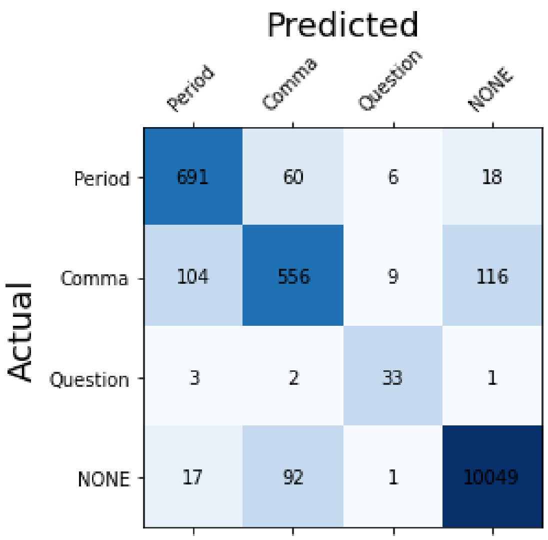

Figure 3. Confusion matrix for T5-base evaluated on the IWSLT 2012 test set.

Error Analysis

Table 5 shows a few examples of prediction errors made by our model. Example 1 and 2 illustrate cases where it is not clear that the model output is in fact wrong. Example 3 shows an unusual sentence that is arguably harder for the model and example 4 shows a mistake in a more common looking text.

Table 5. Examples of model errors. Disparities between the original and the model prediction are marked with square brackets.

	Ground truth	Model output
1	the space they create in the middle creates a new shape[,] the answer to the sum.	the space they create in the middle creates a new shape[.] the answer to the sum.
2	what about bigger numbers? well[] you cannot get much	what about bigger numbers? well[,] you cannot get much
3	school taught you to do math[,] i'm sure[.] it's 16[,] 16[,] 16[,] 48, 4,800, 4,000	school taught you to do math[.] i'm sure[] it's 16[] 16[] 16[.] 48, 4,800, 4000
4	and so we got to the grave and made this, which was hilarious[,] the attention that we got.	and so we got to the grave and made this, which was hilarious[.] the attention that we got.

Conclusion and Future Work

In this work, we demonstrate how seq2seq encoder-decoder transformer models, such as T5, can be used for punctuation restoration. Testing a fine-tuned model on the IWSLT 2012 benchmark, we find its performance comparable to the state-of-the-art.

The flexibility of the seq2seq architecture allows our model to seamlessly predict more than a single punctuation mark in a row, such as an ellipsis (...), an interrobang (!?), or when trying to restore additional types of punctuation marks, such as quotation marks or parentheses, as in (.”). We believe that a benchmark that covers these types of punctuation, would be useful to evaluate punctuation restoration models but leave that as future work.

Acknowledgments

I’d like to thank my teacher, Ms. Duran, for supporting and guiding me in this work.

References

Alexei Baevski and Henry Zhou and Abdelrahman Mohamed and Michael Auli (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. CoRR, abs/2006.11477. https://doi.org/10.21437/interspeech.2021-717

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., & Dahlgren, N. L. (1993). DARPA TIMIT: https://doi.org/10.6028/nist.ir.4930

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2015.7178964

Tündik, M. Á., Szaszák, G., Gosztolya, G., & Beke, A. (2018). User-centric Evaluation of Automatic Punctuation in ASR Closed Captioning. Interspeech 2018. https://doi.org/10.21437/interspeech.2018-1352

Tilk, O., & Alumäe, T. (2016). Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. Interspeech 2016. https://doi.org/10.21437/interspeech.2016-1517

Tilk, O., & Alumäe, T. (2015). LSTM for punctuation restoration in speech transcripts. Interspeech 2015. https://doi.org/10.21437/interspeech.2015-240

Attila Nagy and Bence Bial and Judit Ács (2021). Automatic punctuation restoration with BERT models. CoRR, abs/2101.07343.

Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR, abs/1810.04805. https://doi.org/10.18653/v1/N19-1423

Courtland, M., Faulkner, A., & McElvain, G. (2020). Efficient Automatic Punctuation Restoration Using Bidirectional Transformers with Robust Inference. In Proceedings of the 17th International Conference on Spoken Language Translation (pp. 272–279). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.iwslt-1.33

Kolár, J., Svec, J., & Psutka, J. (2004). Automatic punctuation annotation in czech broadcast news speech.

Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. CoRR, abs/1910.10683.

Noam Shazeer and Mitchell Stern (2018). Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. CoRR, abs/1804.04235.

Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, & Blake Hechtman. (2018). Mesh-TensorFlow: Deep Learning for Supercomputers.

Thomas Wolf and Lysandre Debut and Victor Sanh and Julien Chaumond and Clement Delangue and Anthony Moi and Pierric Cistac and Tim Rault and Rémi Louf and Morgan Funtowicz and Jamie Brew (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. CoRR, abs/1910.03771.

Gravano, A., Jansche, M., & Bacchiani, M. (2009). Restoring punctuation and capitalization in transcribed speech. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4741-4744).
https://doi.org/10.1109/ICASSP.2009.4960690

Yinhan Liu and Myle Ott and Naman Goyal and Jingfei Du and Mandar Joshi and Danqi Chen and Omer Levy and Mike Lewis and Luke Zettlemoyer and Veselin Stoyanov (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR, abs/1907.11692.

Alam, F. (2020). Punctuation Restoration using Transformer Models for High-and Low-Resource Languages. In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020) (pp. 132–142). Association for Computational Linguistics.
https://doi.org/10.18653/v1/2020.wnut-1.18

Stephan Peitz, Markus Freitag, Arne Mauser, & H. Ney (2011). Modeling punctuation prediction as machine translation. IWSLT.

Vāravs, Andris & Salimbajevs, Askars. (2018). Restoring Punctuation and Capitalization Using Transformer Models: 6th International Conference, SLSP 2018, Mons, Belgium, October 15–16, 2018, Proceedings.
https://doi.org/10.1007/978-3-030-00810-9_9