

XVII. Magyar Számítógépes Nyelvészeti Konferencia Szeged, 2021. január 28–29.

Automatic punctuation restoration with BERT models

Attila Nagy¹, Bence Bial¹, Judit Ács¹¹Department of Automation and Applied Informatics
Budapest University of Technology and Economics

Abstract. We present an approach for automatic punctuation restoration with BERT models for English and Hungarian. For English, we conduct our experiments on Ted Talks, a commonly used benchmark for punctuation restoration, while for Hungarian we evaluate our models on the Szeged Treebank dataset. Our best models achieve a macro-averaged F_1 -score of 79.8 in English and 82.2 in Hungarian. Our code is publicly available¹.

1 Introduction

Automatic Speech Recognition (ASR) systems typically output unsegmented transcripts without punctuation. Restoring punctuations is an important step in processing transcribed speech. Tündik et al. (2018) showed that the absence of punctuations in transcripts affect readability as much as a significant word error rate. Downstream tasks such as neural machine translation (Vandeghinste et al., 2018), sentiment analysis (Cureg et al., 2019) and information extraction (Makhoul et al., 2005) also benefit from having clausal boundaries. In this paper we present models for automatic punctuation restoration for English and Hungarian. Our work is based on a state-of-the-art model proposed by (Courtland et al., 2020), which uses pretrained contextualized language models (Devlin et al., 2018).

Our contributions are twofold. First, we present the implementation of an automatic punctuation model based on a state-of-the-art model (Courtland et al., 2020) and evaluate it on an English benchmark dataset. Second, using the same architecture we propose an automatic punctuator for Hungarian trained on the Szeged Treebank (Csendes et al., 2005). To the best of our knowledge our work is the first punctuation restoration attempt that uses BERT on Hungarian data.

2 Related Work

Systems that are most efficient at restoring punctuations usually exploit both prosodic and lexical features with hybrid models (Szaszák and Tündik, 2019; Garg et al., 2018; Żelasko et al., 2018). Up until the appearance of BERT-like

¹ <https://github.com/attilanagy234/neural-punctuator>

models, lexical features were primarily processed by recurrent neural networks (Vandeghinste et al., 2018; Tündik et al., 2017; Kim, 2019; Tilk and Alumäe, 2016; Salloum et al., 2017), while more recent approaches use the transformer (Vaswani et al., 2017) architecture (Chen et al., 2020; Nguyen et al., 2019; Cai and Wang, 2019). The current state-of-the-art method by Courtland et al. (2020) is a pretrained BERT, which aggregates multiple predictions for the same token, resulting in higher accuracy and significant parallelism.

3 Methodology

We train models for Hungarian and English. For English we rely on the widely used IWSLT 2012 Ted Talks dataset (Federico et al., 2012) benchmark dataset. Due to the lack of such datasets for Hungarian, we generate it from the Szeged Treebank (Csendes et al., 2005). We preprocess the Szeged Treebank such that it structures similarly to the output of an ASR system. Then with the presented methods we attempt to reconstruct the original and punctuated gold standard corpus.

3.1 Problem formulation

We formulate the problem of punctuation restoration as a sequence labeling task with four target classes: *EMPTY*, *COMMA*, *PERIOD*, and *QUESTION*. We do not include other punctuation marks as their frequency is very low in both datasets. For this reason, we apply a conversion in cases where it is semantically reasonable: we convert exclamation marks and semicolons to periods and colons and quotation marks to commas. We remove double and intra-word hyphens, however, if they are encapsulated between white spaces, we convert them to commas. Other punctuation marks are disregarded during our experiments. As tokenizers occasionally split words to multiple tokens, we apply masking on tokens, which do not mark a word ending. These preprocessing steps and the corresponding output labels are shown in Table 1.

3.2 Datasets

IWSLT 2012 Ted Talks dataset We use the IWSLT 2012 Ted Talks dataset (Federico et al., 2012) for English. IWSLT is a common benchmark for automatic punctuation. It contains 1066 unique transcripts of Ted talks with a total number of 2.46M words in the corpus. We lowercase the data and we convert consecutive spaces into single spaces. We also remove spaces before commas. We use the original train, validation and test sets from the IWSLT 2012 competition. The overall data distributions of the IWSLT Ted Talk dataset is summarized in Table 2.

Original	Tyranosaurus asked: kill me?
Preprocessed	tyranosaurus asked, kill me?
Tokenized	ty ##rano ##saurus asked kill me
Output	- - EMP COM EMP Q
Original	Not enough, – said the co-pilot – ...
Preprocessed	not enough, said the co pilot,
Tokenized	not enough said the co pilot
Output	EMP COM EMP EMP EMP COM

Table 1: An example input sentence and the following processing steps in our setup.

	Train	Validation	Test
PERIOD	139,619	909	1,100
COMMA	188,165	1,225	1,120
QUESTION	10,215	71	46
EMPTY	2,001,462	15,141	16,208

Table 2: Label distributions of the IWSLT Ted talk dataset.

Szeged Treebank We use the Szeged Treebank dataset (Csendes et al., 2005) for Hungarian. This dataset is the largest gold standard treebank in Hungarian. It covers a wide variety of domains such as fiction, news articles, and legal text. As these subcorpora have very different distributions in terms of punctuations, we merge them and shuffle the sentences. We then split the dataset into train, validation and test sets. This introduces a bias in the prediction of periods as it is easier for the model to correctly predict sentence boundaries by recognizing context change between adjacent sentences but it also provides a more-balanced distribution of punctuation classes across the train, validation and test sets. The label distribution is listed in Table 3.

3.3 Architecture

Our model is illustrated in Figure 3. We base our model on pretrained BERT models. BERT is a contextual language model with multiple transformer layers and hundreds of millions trainable parameters trained on a massive English corpora with the masked language modeling objective. Several variations of pre-trained weights were released. We use BERT-base cased and uncased for English as well as Albert (Lan et al., 2019), a smaller version of BERT. BERT also has

	Train	Validation	Test
PERIOD	81,168	9,218	3,370
COMMA	120,027	13,781	4,885
QUESTION	1,808	198	75
EMPTY	885,451	101,637	36,095

Table 3: Overall data distributions of the Szeged Treebank dataset.

a multilingual version, *mBERT* that supports Hungarian along with 100 other languages. We use mBERT and the recently released Hungarian-only BERT, *huBERT* (Nemeskey, 2020) for Hungarian. These models all apply wordpiece tokenization with their own predefined WordPiece vocabulary. They then generate continuous representations for every wordpiece. Our model adds a two-layer multilayer perceptron on top of these representation with 1568 hidden dimension, ReLU activation and an output layer, and finally a softmax layer that produces a distribution over the labels. We also apply dropout with a probability of 0.2 before and after the first linear layer. Similarly to Courtland et al. (2020), we apply a sliding window over the input data, generate multiple predictions for each token and then aggregate the probabilities for each position by taking the label-wise mean and thus output the most probable label. The process is illustrated in Figure 1 and 2.

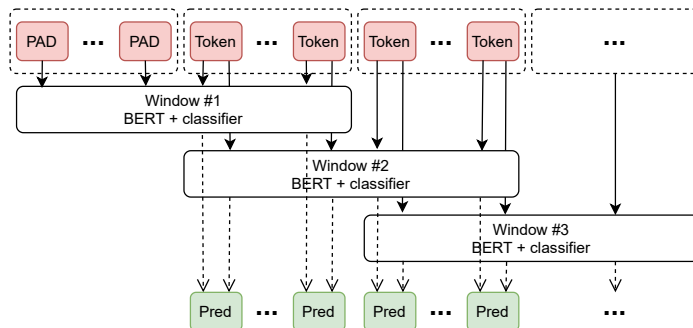


Fig. 1: The process of generating multiple predictions for a token. Although BERT always receives sequences of 512, we sample consecutive sequences from the corpora such that they overlap, thus resulting in multiple predictions for the same token. The extent of the overlap and therefore the number of predictions for a token depend on the offset between the windows. Note that padding is necessary in the beginning to ensure that all tokens have the same amount of predictions.

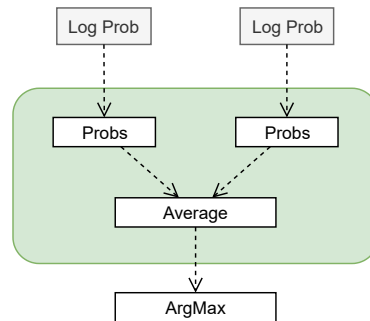


Fig. 2: The final prediction is computed by first aggregating all punctuation probability distributions for each token by taking their class-wise averages and then selecting the highest probability.

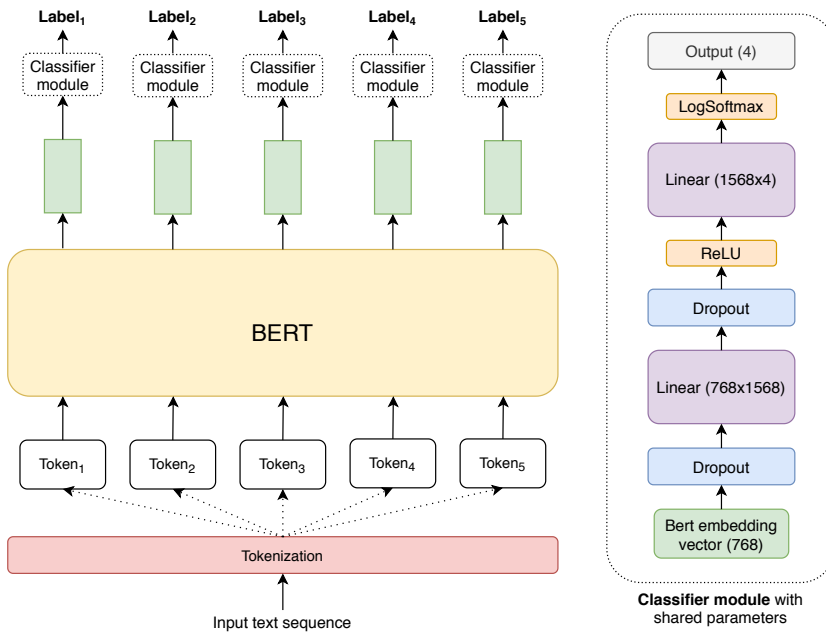


Fig. 3: The complete architecture used for punctuation restoration.

3.4 Training

We train all models with identical hyperparameters. We perform gradient descent using the AdamW optimizer (Loshchilov and Hutter, 2017) with the learning rate set to $3 * 10^{-5}$ for BERT and 10^{-4} for the classifier on top. We apply gradient clipping of 1.5 and a learning rate warm up of 300 steps using a linear scheduler. We select negative log likelihood as the loss function. The tokenizer modules often split single words to multiple subwords. For this task we only need to predict punctuations after words (between white spaces). We mask the loss function for every other subword. It is a common practice to intermittently freeze and unfreeze the weights of the transformer model, while training the fine-tuning linear layers situated at the top of the whole architecture. We found that it is best to have the transformer model unfrozen from the very first epoch and therefore update its parameters along with the linear layers. We trained the models for 12 epochs with a batch size of 4 and applied early stopping based on the validation set. We used the validation set to tune the sliding window step size, that is responsible for getting multiple predictions for a single token. All experiments were performed using a single Nvidia GTX 1070 GPU with one epoch taking 10 minutes. Our longest training lasted for 2 hours.

4 Results

All models are evaluated using macro F_1 -score (F) over the 4 classes. Similarly to Courtland et al., our work is focused on the performance of punctuation marks and as *EMPTY* labels constitute 85% of all labels, we report the overall F_1 -score without *EMPTY*. We evaluated both cased and uncased variations of BERT and generally we have found that the uncased model is better than its cased variant for this task. This was an expected conclusion, as we lowercased the entire corpus with the purpose of eliminating bias around the prediction of periods. For all setups, we selected the best performing models on the validation set by loss and by macro F_1 -score and evaluated them independently on the test set. On the Ted Talks dataset, our best performing model was an uncased variation of BERT that achieved on par performance with the current state-of-the-art model (Courtland et al., 2020), having a slightly worse macro F_1 -score of 79.8 (0.8 absolute and 0.9975% relative difference) with 10 epochs of training and 64 predictions/token. All results on the Ted Talks dataset are summarized in Table 4.

On the Szeged Treebank dataset, we evaluate the multilingual variants of BERT and the recently released Hubert model. We find that Hubert performs significantly better (82.2 macro F_1 -score) than the best multilingual model with an absolute and relative difference of 12.2 and 14.84% respectively on macro F_1 -score. We trained the best Hubert model for 3 epochs and used 8 predictions/token. All results on the Szeged Treebank dataset are summarized in Table 5.

We also examined the effect of using multiple predictions for a token. The changes we see in macro F_1 -score on the validation set with regard to the number

Models	Comma			Period			Question			Overall		
	P	R	F	P	R	F	P	R	F	P	R	F
BERT-base (Courtland et al., 2020)	72.8	70.8	71.8	81.9	86.6	84.2	80.8	91.3	85.7	78.5	82.9	80.6
Albert-base (Courtland et al., 2020)	69.4	69.3	69.4	80.9	84.5	82.7	76.7	71.7	74.2	75.7	75.2	75.4
BERT-base-uncased (by loss)	59.0	80.2	68	83.0	83.6	83.3	87.8	83.7	85.7	76.6	82.5	79.0
BERT-base-uncased (by F_1 -score)	58.4	80.7	67.8	84.2	83.8	84.0	84.8	90.7	87.6	75.8	85.1	79.8
BERT-base-cased (by loss)	57.3	73.9	64.5	75.9	87.9	81.4	77.1	84.1	80.4	70.1	81.9	75.5
BERT-base-cased (by F_1 -score)	59.1	78.5	67.5	79.6	81.6	80.6	76.9	88.9	82.5	71.9	83.0	76.8
Albert-base (by loss)	55.3	74.8	63.6	76.8	87.9	82.0	70.6	83.7	76.6	67.6	82.1	74.1
Albert-base (by F_1 -score)	56.5	80.3	66.3	80.7	80.8	80.8	80.4	84.1	82.2	72.5	81.7	76.4

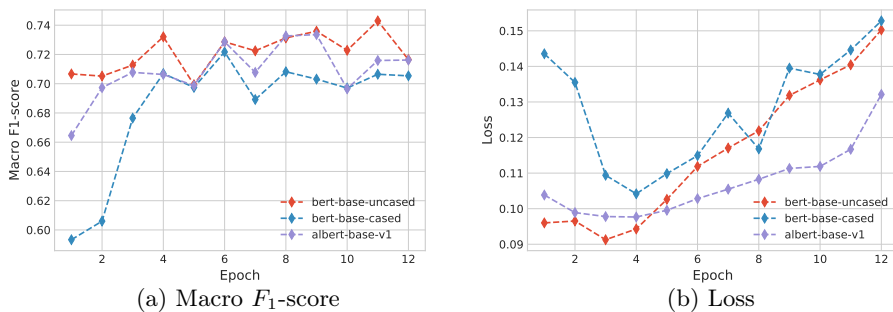
 Table 4: Precision, recall and F_1 -score values on the Ted Talk dataset.


Fig. 4: Metrics on the validation set over epochs during training on the IWSLT Ted Talk dataset

Models	Comma			Period			Question			Overall		
	P	R	F	P	R	F	P	R	F	P	R	F
BERT-base-multilingual-uncased (by loss)	82.3	79.3	80.8	79.6	88.3	83.8	43.2	21.3	28.6	68.4	63.0	64.4
BERT-base-multilingual-uncased (by F_1 -score)	82.9	79.4	81.1	80.1	88.4	84.0	51.4	24.0	32.7	71.5	63.9	66.0
BERT-base-multilingual-cased (by loss)	81.3	79.3	80.3	82.4	83.2	82.8	51.6	21.3	30.2	71.8	61.3	64.4
BERT-base-multilingual-cased (by F_1 -score)	83.6	78.8	81.1	81.7	85.5	83.6	61.4	36.0	45.4	75.6	66.8	70.0
Hubert (by loss and F_1 -score)	84.4	87.3	85.8	89.0	93.1	91.0	73.5	66.7	69.9	82.3	82.4	82.2

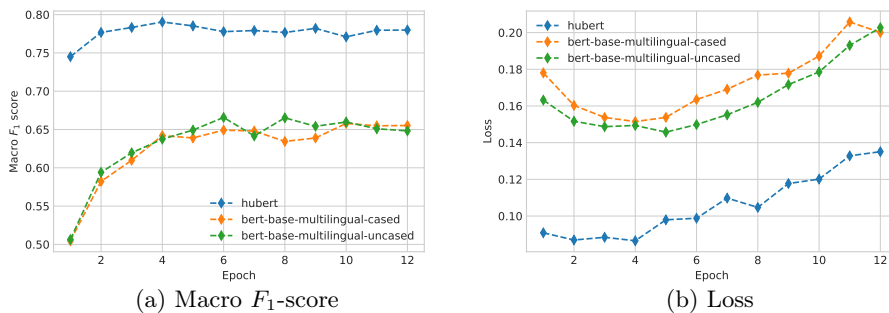
 Table 5: Precision, recall and F_1 -score values on the Szeged Treebank dataset.


Fig. 5: Metrics on the validation set over epochs during training on the Szeged Treebank dataset.

of predictions per token are shown in Figure 6. The best models were evaluated on the test set and we found that having multiple predictions per token increased the F_1 -score by 5% in English and 2.4% in Hungarian.

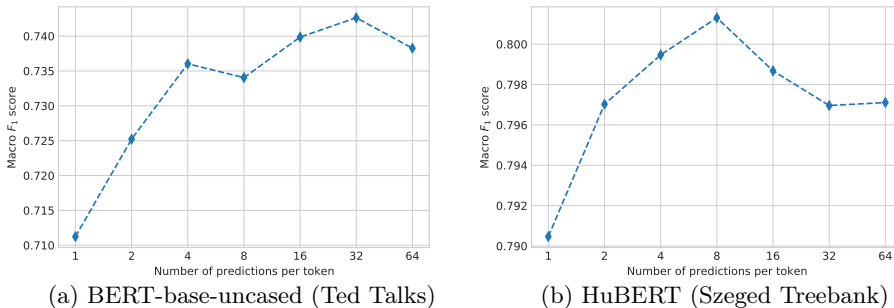


Fig. 6: Effect of the number of predictions per token on the overall F_1 -score, computed on the validation dataset.

5 Conclusion

We presented an automatic punctuation restoration model based on BERT for English and Hungarian. For English we reimplemented a state-of-the-art model and evaluated it on the IWSLT Ted Talks dataset. Our best model achieved comparable results with current state-of-the-art on the benchmark dataset. For Hungarian we generated training data by converting the Szeged Treebank into an ASR-like format and presented BERT-like models that solve the task of punctuation restoration efficiently, with our best model Hubert achieving a macro F_1 -score of 82.2.

Bibliography

- Cai, Y., Wang, D.: Question mark prediction by bert. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). pp. 363–367. IEEE (2019)
- Chen, Q., Chen, M., Li, B., Wang, W.: Controllable time-delay transformer for real-time punctuation prediction and disfluency detection. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8069–8073. IEEE (2020)
- Courtland, M., Faulkner, A., McElvain, G.: Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In: Proceedings of the 17th International Conference on Spoken Language Translation. pp. 272–279 (2020)

- Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: International Conference on Text, Speech and Dialogue. pp. 123–131. Springer (2005)
- Cureg, M.Q., De La Cruz, J.A.D., Solomon, J.C.A., Saharkhiz, A.T., Balan, A.K.D., Samonte, M.J.C.: Sentiment analysis on tweets with punctuations, emoticons, and negations. In: Proceedings of the 2019 2nd International Conference on Information Science and Systems. pp. 266–270 (2019)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- Federico, M., Cettolo, M., Bentivogli, L., Michael, P., Sebastian, S.: Overview of the iwslt 2012 evaluation campaign. In: IWSLT-International Workshop on Spoken Language Translation. pp. 12–33 (2012)
- Garg, B., et al.: Analysis of punctuation prediction models for automated transcript generation in mooc videos. In: 2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 19–26. IEEE (2018)
- Kim, S.: Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7280–7284. IEEE (2019)
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
- Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
- Makhoul, J., Baron, A., Bulyko, I., Nguyen, L., Ramshaw, L., Stallard, D., Schwartz, R., Xiang, B.: The effects of speech recognition and punctuation on information extraction performance. In: Ninth European Conference on Speech Communication and Technology (2005)
- Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)
- Nguyen, B., Nguyen, V.B.H., Nguyen, H., Phuong, P.N., Nguyen, T.L., Do, Q.T., Mai, L.C.: Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. In: 2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA). pp. 1–5. IEEE (2019)
- Salloum, W., Finley, G., Edwards, E., Miller, M., Suendermann-Oeft, D.: Deep learning for punctuation restoration in medical reports. In: BioNLP 2017. pp. 159–164 (2017)
- Szaszák, G., Tündik, M.Á.: Leveraging a character, word and prosody triplet for an asr error robust and agglutination friendly punctuation approach. In: INTERSPEECH. pp. 2988–2992 (2019)
- Tilk, O., Alumäe, T.: Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In: Interspeech. pp. 3047–3051 (2016)

- Tündik, M.A., Szaszák, G., Gosztolya, G., Beke, A.: User-centric evaluation of automatic punctuation in asr closed captioning (2018)
- Tündik, M.Á., Tarján, B., Szaszák, G.: A bilingual comparison of maxent-and rnn-based punctuation restoration in speech transcripts. In: 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). pp. 000121–000126. IEEE (2017)
- Vandeghinste, V., Verwimp, L., Pelemans, J., Wambacq, P.: A comparison of different punctuation prediction approaches in a translation context. Proceedings EAMT (2018)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
- Żelasko, P., Szymański, P., Mizgajski, J., Szymczak, A., Carmiel, Y., Dehak, N.: Punctuation prediction model for conversational speech. arXiv preprint arXiv:1807.00543 (2018)