

PunKtuator: A Multilingual Punctuation Restoration System for Spoken and Written Text

Varnith Chordia

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA - 94034
vchordia@parc.com

Abstract

Text transcripts without punctuation or sentence boundaries are hard to comprehend for both humans and machines. Punctuation marks play a vital role by providing meaning to the sentence and incorrect use or placement of punctuation marks can often alter it. This can impact downstream tasks such as language translation and understanding, pronoun resolution, text summarization, etc. for humans and machines. An automated punctuation restoration (APR) system with minimal human intervention can improve comprehension of text and help users write better. In this paper we describe a multitask modeling approach as a system to restore punctuation in multiple high resource – Germanic (English and German), Romanic (French)– and low resource languages – Indo-Aryan (Hindi) Dravidian (Tamil) – that does not require extensive knowledge of grammar or syntax of a given language for both spoken and written form of text. For German language and the given Indic based languages this is the first towards restoring punctuation and can serve as a baseline for future work.

1 Introduction

Automatic speech recognition (ASR) has become ubiquitous these days and has wide applications in business and personal life. One of the drawbacks of ASR is it produces an unpunctuated stream of text. Restoring punctuation manually is a time-consuming task. Apart from spoken text a large amount of written text online - blogs, articles, social media, etc. - sometimes lack the appropriate punctuation marks due to human inconsistencies, which can alter the meaning of text. An APR system designed with an understanding of ASR and written forms of text can help resolve these issues. Transcriptions passed to an APR system, can improve the following machine learning tasks such

as machine translation, conversational agents, coreference resolution, etc. Further it can be used as an unsupervised auxiliary or pretext task, for training large scale transformer language models, as it would require understanding about global structure of the text.

Prior punctuation restoration methods have mostly been solved using lexical features, prosodic features or combination of both. Due to large availability of text data, majority of the methods have focused on using lexical features. Early methods (Christensen et al., 2001) used Hidden Markov Models (HMM) to model punctuation using acoustic features such as pause duration, pitch and intensity. Though the acoustic based models perform well on ASR system, they can perform better when combined with textual data. Liu et al. (2006); Batista et al. (2007); Kolář and Lamel (2012) proposed various methods that combined lexical features along with prosodic information thereby improving APR tasks. Tilk and Alumäe (2015, 2016) proposed unidirectional and bidirectional Long Short Term Memory (Bi-LSTM) based punctuation prediction model which did not require extensive feature engineering. Though the above method considered the long distant token dependencies, it ignored label dependencies. To address label dependencies (Klejch et al., 2017) made use of recurrent neural networks for sequence to sequence mapping using an encoder-decoder architecture. Recently the use of transformer based approaches combination of speech and pre-trained word embeddings have achieved state of art performance on IWSLT datasets (spoken transcripts from TED talks for ASR tasks, but often used as benchmark for comparison of punctuation restoration models). Yi et al. (2020) used pretrained BERT (Devlin et al., 2018) that is used to perform adversarial multi-task learning to restore punctuation. Alam et al. (2020) explored various transformer model architectures

Language	Total	Other (O)	Period	Comma	Question
English	48,334,765	43,855,115 (90.7%)	1,914,198 (3.96%)	2,492,124 (5.15%)	55,123 (0.11%)
French	22,315,779	20,223,881 (90.6%)	1,232,798 (5.5%)	815,123 (3.6%)	30,379 (0.11%)
German	41,269,527	36,438,630 (88.2%)	1,875,766 (4.54%)	2,816,425 (6.82%)	60216 (0.14%)
Hindi	3,388,500	3,135,290 (92.5%)	127,101 (3.75%)	100,710 (2.97%)	1246 (0.03%)
Tamil	906,271	791,6320 (87.8%)	58,963 (6.50%)	53,198 (5.86%)	1,496 (0.09%)

Table 1: Distribution of the High resource and long resource language datasets. The top-3 languages in the table are considered high Resource, while the bottom 2 are low resource languages

and used an augmentation strategy to make models more robust to ASR errors.

Though most approaches have shown considerable improvement in overcoming some of the challenges faced in terms of modeling and achieving the state of performance in spoken language transcripts in English, there are the following limitations:

- Restoring punctuation varies in spoken and written text due to differences in rules of writing and speaking. The frequent use of personal pronouns, colloquial words and usage of direct speech often results in more varied use of punctuation in spoken text as compared to written text. This often affects readability for humans and machines.
- Though there has been some research (Tilk and Alumäe, 2016; Kolář and Lamel, 2012; Alam et al., 2020) that has focused on developing non-english APR system, extensive research and baseline results have not been studied for other languages.

To overcome some of the challenges, we make the following contributions:

- We implemented a multi-task multilingual punctuation restoration model. Our technique implements punctuation restoration task as sequence labeling task, which is jointly trained with language classifiers and text mode classification ('Spoken' and 'Written'). We use the proposed technique to build two multilingual models for high resource and low resource languages, thereby reducing the dependency of multiple monolingual language models.

- We developed a web browser extension that can help multilingual spoken and written users to punctuate transcripts as a post-processing step. We have made a demo of the web extension available online.¹
- We prepared training and test datasets and evaluated the performance of our proposed model. Further to evaluate the generalization of the model we evaluated across the benchmark IWSLT reference dataset. The code and models have been made publicly available.²

2 Punctuation restoration system

2.1 Data Gathering

Due to varying set of language data, we segregated the data sources according to the languages, which we gathered for spoken and written text. For Written text we considered data from news web sources.

2.1.1 High Resource Languages

For high resource European languages, we considered a parallel sentence corpus known as the 'EUROPART' corpus (Vanmassenhove and Hardmeier, 2018) for spoken text. This corpus is a collection of speeches made in the proceedings of European parliament from 1996 to 2012, transcribed as text. To gather written text we used news articles from Alexa's top-25 ranked news sources. These were publicly available³ for every language.

¹<https://youtu.be/9FdkuENPhuY>

²https://github.com/VarnithChordia/Multilingual_Punctuation_restoration

³<https://webhose.io/>

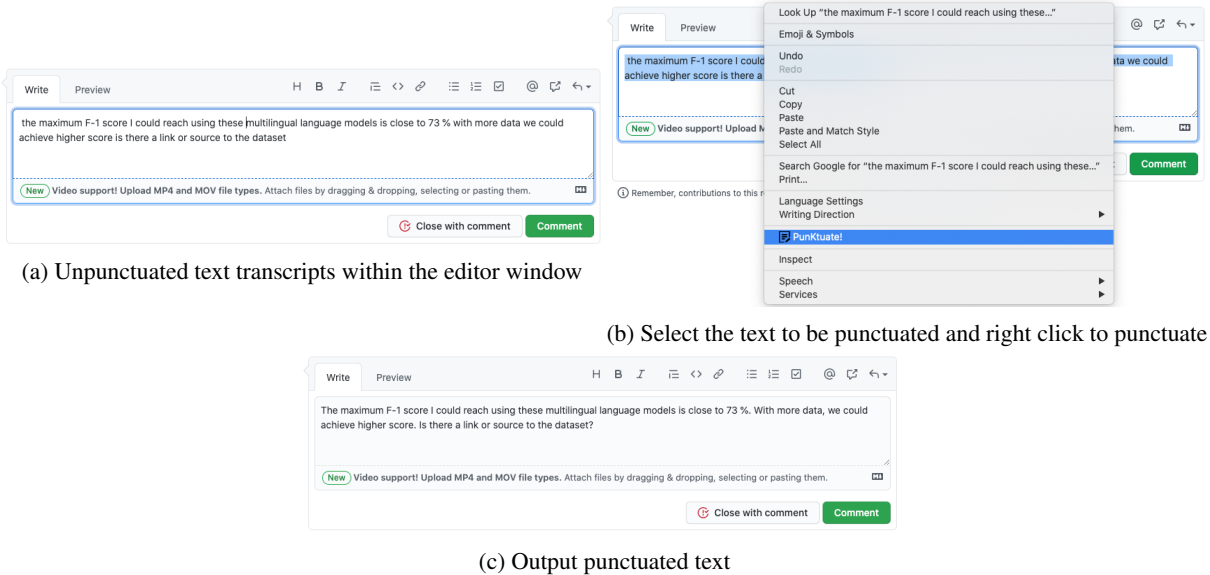


Figure 1: Example of punctuation via web extension.
Source: www.github.com

2.1.2 Low Resource Languages

Due to lack of language resources available for indic languages for APR, we gather publicly released datasets. For Spoken text we used the Indian Prime Minister’s address to the nation. These corpora manually translated into several Indian languages. Written text was obtained from [Siripragada et al. \(2020\)](#) who crawled articles released from the Press Information Bureau (PIB), an Indian government agency that provides information to news media.

2.2 Annotation

Due to lack of readily available annotated datasets and large size corpora, we used an automated approach to label the data. We analyzed languages and selected the three most common punctuation – ‘PERIOD’, ‘COMMA’ and ‘QUESTION MARK’ – that occurred across the languages for training our model. This was done to improve the readability of text so that could be easily understood by users, one of the goals of the system. Since we treat our task as a sequence labeling task, we annotated every word in the sequence according to the punctuation following it. We achieved this by tokenizing the input text into a stream of word tokens and punctuation tokens. We converted this into a set of pairs of (token, punctuation) where punctuation is the null punctuation (‘O’), if there was no punctuation mark following in the text. To make our data set more diverse and training more

robust, we ended sentences (10%) a few tokens before the ‘PERIOD’ tag and labeled the final token as ‘EOS’ (end of sentence). Further we converted all our text to lowercase to remove any signal while training the language model. The distribution of the labels can be seen in table 1.

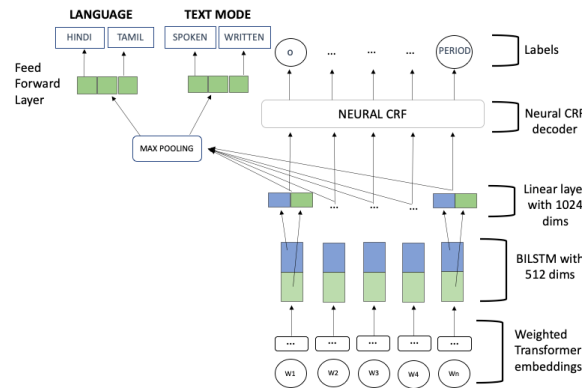


Figure 2: Joint Punctuation model on indic languages for spoken and written text.
 Looks better when Zoomed.

2.3 Joint Multilingual Model-Architecture

The model consists of four main sub-parts as observed in Figure 2 – (i) Transformer Language Model (ii) BILSTM (iii) Neural Conditional Random Field (NCRF++) and (iv) Language and Text Mode classifiers. We use a pretrained transformer language model to generate word/sub-word embeddings, but do not fine tune this model to our specific task. A BILSTM on top of the transformer

Model	Language	TextMode	Period	Comma	Question	Overall
Joint - BILSTM NCRF + FastText	English	Spoken	92.3	66.7	65.2	74.8
		Written	83.3	57.6	32.5	57.7
	French	Spoken	88.9	64.2	30.8	61.3
		Written	77.0	56.7	33.9	55.9
	German	Spoken	93.2	88.6	48.0	76.6
		Written	77.3	71.0	31.0	64.1
Joint - XLMLRoberta NCRF	English	Spoken	92.8	79.9	86.7	86.5
		Written	90.4	75.9	80.7	82.3
	French	Spoken	92.8	80.9	82.7	85.5
		Written	83.8	66.2	75.1	66.2
	German	Spoken	95.8	93.8	85.7	91.8
		Written	91.8	88.8	69.7	83.4
Joint - Multilingual-BERT NCRF	English	Spoken	95.8	80.6	92.8	90.1
		Written	96.0	82.7	79.7	86.4
	French	Spoken	94.8	80.8	88.1	89.2
		Written	93.5	78.5	73.0	81.7
	German	Spoken	97.0	95.0	90.7	94.2
		Written	96.0	74.3	90.0	86.8
Punctuation Restoration (Alam et al., 2020)	English	Spoken	80.8	75.0	78.7	78.1

Table 2: Results on High resource languages. The values in bold indicate the best performing model.

model is used to model token dependencies better, from forward and backward directions. NCRF (Yang and Zhang, 2018) relies on learning the high level features from the deep neural network and passes this information to a linear CRF layer for inference, which helps manage label dependencies. This architecture sequential in nature, is trained for APR task. The output sequence representation from the BILSTM is passed through a max pooling layer, the result of which passed through linear feed forward layer for language and text mode classification. We jointly trained our sequential language model, along with the classifiers.

2.4 Web Extension

We created a web extension that can be used to punctuate text within the text editors on web pages. It lets users to select text which could range from a few words to large paragraphs to entire documents to punctuate. The text does not have to be non punctuated as the system removes punctuation as a preprocessing step and punctuates again. The steps to punctuate are shown in Fig 1.

3 Experiments

3.1 Experimental setup

We used the pretrained transformer model and specific tokenizers available on HuggingFace⁴. The model architecture consists of the 12 hidden layer encoder, which is used to produce the embeddings. We used an optimized weighting technique (Peters et al., 2018) to sum all the hidden layers rather than use a common practice of using one single layer to generate embeddings. This showed an improvement in performance as seen under ablation studies in table 5. The weighting method is as defined:

$$O_i = \gamma \sum_{j=0}^{L-1} S_j H_j \quad (1)$$

where

- H_j is a trainable task weight for the j^{th} layer. γ is another task trainable task parameter that aids the optimization process
- S_j is the normalized embedding output from the j^{th} hidden layer of the transformer.
- O_j is the output vector.

⁴<https://huggingface.co/>

- L is the number of hidden layers.

To train the proposed model, we used a maximum sequence length of 505. We use a subword tokenization technique - sentence piece model (Kudo and Richardson, 2018) - which might result in token length exceeding the maximum sequence length, in such cases we exclude the tokens and start a new paragraph. For sequences less than the specified max sequence length, we pad the sequences to the maximum sequence length and mask the padded sequence to avoid performing attention on it. We used a batch size of 32, grouping similar sequence length prior to padding that enhances the speed while training the model. We do not fine tune the transformer model, but use it to embed the input text. A BILSTM stacked on top of the transformer model, is set to a dimension of 512, the layers are initialized with a uniform distribution in the range of (-.003, .003). A Neural CRF layer is trained with a maximum log-likelihood loss. Viterbi algorithm is used to search for the label sequence with the highest probability during decoding. The entire model was trained with an Adam optimization algorithm with a learning rate close to $1e-4$ over 10 epochs. The proposed multitask network was trained via a dynamically weighted averaging (DWA) technique to balance each task. Thereby not allowing one task to dominate over the other or negatively impact the performance of the other. This approach was proposed and utilized for training a multi-task computer vision network (Liu et al., 2019), we followed a similar approach and implemented this on language processing task to show overall improvement in performance. Similar to Gradnorm (Chen et al., 2018) which learns to average tasks over time, the DWA method does not use the gradients of network rather uses numeric task loss. The weighting λ_j for task j is defined as:

$$\lambda_j = \frac{K \exp(w_j(n-1)/T)}{\sum_i \exp(w_i(n-1)/T)} \quad (2)$$

where

$$w_j(n-1) = \frac{L_j(n-1)}{L_j(n-2)} \quad (3)$$

L_j is the loss function of each task j , so w_j is the ratio of loss function over the last two epochs. T represents the temperature, which is used to represent the softness of task weighting. A higher value of T represents a more even distribution between the tasks, when T is high enough, the value of λ_j

equals 1. K is the total number of the tasks that we are training for. The overall loss is the sum of the individual task loss averaged over each iteration.

$$L_{ovrl} = \frac{\lambda_1 L_{pr} + \lambda_2 L_{lc} + \lambda_3 L_{tm}}{batchsize} \quad (4)$$

where L_{pr} - Maximum Likelihood loss for Punctuation restoration, L_{lc} - Cross Entropy loss for Language Classification and L_{tm} - Cross entropy loss for text mode classification.

3.2 Results

To evaluate the performance of our joint model, we built different multilingual neural models. We split our dataset into two parts — train set (80%), validation set (10%) and test set (10%). The performance for every model was evaluated on test set, after being trained on the train set. We chose F1-score to evaluate the performance of our model. We established a baseline using BILSTM-CRF and pretrained FastText word embeddings (Bojanowski et al., 2017) as features and trained jointly on language and text mode classification tasks. The FastText word embeddings used as features for training are monolingual. To train multilingual models, we developed cross lingual embeddings by aligning monolingual embeddings of different languages along a single dimension using unsupervised techniques (Chen and Cardie, 2018). The parameters and training setup of the baseline was similar to the proposed model, except we used FastText based word embeddings as input features. Further we make comparisons using MBERT and XLM-Roberta as pretrained models. Table 2 shows the performance of the various models on high resource European languages along with their F1 scores. To ensure a fairer comparison, we implemented the trained model by Alam et al. (2020) that achieved state of art performance on IWSLT datasets to evaluate on our test set. The Joint-Multilingual BERT NCRF as proposed in section 2.3 outperforms the other models across spoken and written text for all punctuations. We observe German language performs the best across spoken and written text. The performance of the German language can be attributed to a couple of reasons. In German multiple words can be condensed into a single word. This reduces ambiguity and thus there are fewer decision points for the machine to provide inference on. German is an inflected language i.e the word order changes according to the function in the sentence. Most word orders are

Model	Language	TextMode	Period	Comma	Question	Overall
Joint - BILSTM NCRF + FastText	Hindi	Spoken	84.8	47.0	47.6	59.6
		Written	89.2	34.7	55.2	59.7
	Tamil	Spoken	59.8	40.3	19.4	39.8
		Written	47.2	24.9	14.9	29.0
Joint - XLMLRoberta NCRF	Hindi	Spoken	88.7	67.3	41.1	65.6
		Written	92.3	70.8	43.4	68.5
	Tamil	Spoken	75.9	58.7	20.3	56.6
		Written	70.5	43.6	20.3	44.8
Joint - Multilingual BERT NCRF	Hindi	Spoken	90.6	66.6	68.8	75.3
		Written	93.7	74.9	59.6	76.1
	Tamil	Spoken	85.3	71.8	66.6	74.6
		Written	74.8	50.1	43.1	56.0

Table 3: Results on low resource languages

defined in terms of finite verb (V), in combination with Subject (S), and object (O). In German, this can vary according to independent or dependent clauses. In cases of independent clauses, the main verb must be the second element in the sentence (SVO) and the past participle the final element. Under dependent clauses, the object must be the second element in the sentence (SOV). This may provide an additional signal to model and that can impact its performance.

Models	F1-Score
DRNN-LWMA-pre (Kim, 2019)	68.6
Self-Attention (Yi and Tao, 2019)	72.9
BERT-Adversarial (Yi et al., 2020)	77.8
Joint M-BERT (Our Model)	80.3
XLM-R Augmented (Alam et al., 2020)	82.9

Table 4: Performance of the Joint Model on the IWSLT Ref dataset in comparison with other models. The table indicates the average F1 Scores.

To assess the ability of our model to generalize, we evaluated our best performing model on the reference transcripts of the IWSLT dataset. Even though our model was not trained specifically using these datasets, but was able to outperform on some of the prior state of art models as shown in Table 4. The metrics shown refer to the average F1-score. The performance of our proposed models was carried out on the low resource languages for spoken and written transcripts, which can be observed in

Table 3 . We obtained the best result using the Joint-Multilingual BERT NCRF model. For low resource languages the performance of *Question* is lower than the *Comma* and *Period*, due to lower number of questions in true label set.

3.3 Ablation Studies

Models	HRL	LRL
BILSTM-NCRF	66.3	32.3
M-BERT-NCRF	73.5	41.6
M-BERT-BILSTM	78.3	52.4
M-BERT-BILSTM NCRF W/O Weighting Layers	82.6	63.1
M-BERT-BILSTM NCRF W/O Classification layers	84.8	65.2
Our Model	88.0	70.5

Table 5: Ablation Study on our dataset. HRL - High Resource Languages, LRL - Low resource languages

We experimented with different ablations of the best performing model, as seen in table 5 .

- **BILSTM-NCRF** - We do not consider any embeddings and train a simple BILSTM-NCRF model.
- **MBERT-NCRF** - We removed the BILSTM layer and use only NCRF layer on top of transformers.
- **MBERT-BILSTM** - We remove the NCRF layer and model only the token dependencies.
- **Without weighted layers** - We removed the trainable weighing parameters and considered

only the top layer of the transformer as input to the BILSTM.

- **Without classification layers** - We removed the classification layers and trained the model without any auxiliary information.

4 Conclusion

In this paper we described and implemented a joint modeling approach for restoring punctuation for High and low resource languages across spoken and written text. Joint language model trained with auxiliary language and text mode classification improved the performance of the APR task. We achieved reasonable performance on the benchmark IWSLT datasets without being trained on it. We also presented a web extension that can help multilingual users improve overall readability and coherence of text. Further we present baseline results on indic languages that can be used for future work. We have shown examples of punctuated text that was output from our system in the Appendix section.

5 Acknowledgement

We would like to thank Palo Alto Research Center for providing compute resources and Sebastian Safari for providing valuable help in developing the web extension.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 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)*, pages 132–142, Online. Association for Computational Linguistics.

Fernando Batista, Diamantino Caseiro, Nuno Mamede, and Isabel Trancoso. 2007. Recovering punctuation marks for automatic speech recognition. In *Eighth Annual Conference of the International Speech Communication Association*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Xilun Chen and Claire Cardie. 2018. [Unsupervised multilingual word embeddings](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR.

Heidi Christensen, Yoshihiko Gotoh, and Steve Renals. 2001. Punctuation annotation using statistical prosody models.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Seokhwan Kim. 2019. 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)*, pages 7280–7284. IEEE.

Ondřej Klejch, Peter Bell, and Steve Renals. 2017. Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5700–5704. IEEE.

Jáchym Kolář and Lori Lamel. 2012. Development and evaluation of automatic punctuation for french and english speech-to-text. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880.

Yang Liu, Nitesh V Chawla, Mary P Harper, Elizabeth Shriberg, and Andreas Stolcke. 2006. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech & Language*, 20(4):468–494.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Shashank Siripragada, Jerin Philip, Vinay P Namboodiri, and CV Jawahar. 2020. A multilingual parallel corpora collection effort for indian languages. *arXiv preprint arXiv:2007.07691*.

Ottokar Tilk and Tanel Alumäe. 2015. Lstm for punctuation restoration in speech transcripts. In *Sixteenth annual conference of the international speech communication association*.

- Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech*, pages 3047–3051.
- Eva Vanmassenhove and Christian Hardmeier. 2018. Europarl datasets with demographic speaker information.
- Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. *arXiv preprint arXiv:1806.05626*.
- Jiangyan Yi and J. Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7270–7274.
- Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Cunhang Fan. 2020. Adversarial transfer learning for punctuation restoration. *arXiv preprint arXiv:2004.00248*.

A Example Appendix

We present a few examples of text passed to our system in Table 6 and Figure 3 as seen in the next page. It contains two columns - ‘Input Text’ & ‘Output Text’. The ‘Input Text’ columns consists of unpunctuated examples that was passed to our system, while the ‘Output Text’ column is the punctuated text that was returned. The highlighted colors of punctuation marks indicate whether the punctuation was replaced correctly or not. Green indicates the correct punctuation restored, red indicates the incorrect punctuation mark and yellow indicates the missed punctuation mark.

Input Text	Output Text
japan then laid siege to the syrian penalty area for most of the game but rarely breached the syrian defence oleg shatskiku made sure of the win in injury time hitting an unstoppable left foot shot from just outside the area	Japan then laid siege to the syrian penalty area for most of the game but rarely breached the syrian defence. Oleg shatskiku made sure of the win in injury time hitting an unstoppable left foot shot from just outside the area.
russia's refusal to support emergency supply cuts would effectively and fatally undermine OPEC+'s ability to play the role of oil price stabilizing swing producer says Rapidan Energy's Bob McNally	Russia's refusal to support emergency supply cuts would effectively and fatally undermine OPEC +'s ability to play the role of oil price stabilizing. Swing producer, says Rapidan Energy's Bob McNally.
Romeo Romeo wherefore art thou Romeo	Romeo, Romeo, wherefore art thou Romeo?
sans préjuger de l'efficacité de ce couvre-feu avancé ces données ne sont toutefois pas si facilement lisibles selon les experts suivant l'épidémie de Covid-19 Tout d'abord on manque encore de recul	Sans préjuger de l'efficacité de ce couvre-feu avancé, ces données ne sont toutefois pas si facilement lisibles, selon les experts suivant l'épidémie de Covid-19. Tout d'abord, on manque encore de recul.

Table 6: Examples of automatic punctuation restoration of text in our system for European languages.

Input	Output
आखिरी ओवर का खेल शेष था उससे ठीक पहले ऑस्ट्रेलियाई कप्तान टिम पेन ने हार मान ली भारत ने ऐतिहासिक ड्रॉ खेला यह ड्रॉ किसी जीत से कम नहीं चोट से जूझती हुए भारतीय खिलाड़ियों ने विषम हालातों में मैच बचाया	आखिरी ओवर का खेल शेष था, उससे ठीक पहले ऑस्ट्रेलियाई कप्तान टिम पेन ने हार मान ली। भारत ने ऐतिहासिक ड्रॉ खेला। यह ड्रॉ किसी जीत से कम नहीं। चोट से जूझती हुए भारतीय खिलाड़ियों ने विषम हालातों में मैच बचाया।
கோவிஷ்ட்டு கோவாக்சினை யார் யாருக்கு எந்த தடுப்பூசிகளை போடுவது என்பது பற்றி இதுவரை முடிவு செய்யவில்லை தடுப்பூசி தொடர்பாக மத்திய அரசு தொடர்ந்து வழிகாட்டு நெறிமுறைகள் வழங்கி வருகிறது புனேயில் இருந்து காலை 10:30 மணிக்கு கொரோனா தடுப்பு மருந்து சென்னை விமான நிலையம் வருகிறது	கோவிஷ்ட்டு, கோவாக்சினை யார் யாருக்கு எந்த தடுப்பூசிகளை போடுவது என்பது பற்றி இதுவரை முடிவு செய்யவில்லை. தடுப்பூசி தொடர்பாக மத்திய அரசு தொடர்ந்து வழிகாட்டு நெறிமுறைகள் வழங்கி வருகிறது. புனேயில், இருந்து காலை 10:30 மணிக்கு கொரோனா தடுப்பு மருந்து சென்னை விமான நிலையம் வருகிறது.

Figure 3: Indic language example