UZH OnPoint at Swisstext-2021: Sentence End and Punctuation Prediction in NLG Text Through Ensembling of Different Transformers

Andrianos Michail* **Silvan Wehrli*** **Terézia Bucková**
University of Zurich
{andrianos.michail, silvan.wehrli, terezia.buckova}@uzh.ch

Abstract

This paper presents our solutions for the SwissText 2021 shared task "Sentence End and Punctuation Prediction in NLG Text". We engaged with both subtasks (i.e., sentence end detection and full-punctuation prediction) and built systems for English, German, French and Italian. To tackle the punctuation prediction problem, we ensemble multiple differently trained Transformer models (BERT, CamemBERT, Electra, Longformer, MPNet, XLM-RoBERTa, XLNet) and leverage their results using a sliding window method during inference time. As a result, we achieve an F1 score of the positive class of 0.94 for English, 0.96 for German, 0.93 for French, and 0.93 for Italian for the subtask 1 "sentence end detection" on the respective test sets. Furthermore, Macro F1 results on test sets for subtask 2 "full-punctuation prediction" for English, German, French and Italian are 0.78, 0.81, 0.78, 0.76 respectively.

1 Introduction

Transcribed or translated texts often contain erroneous punctuation. Correct punctuation, however, is crucial for human understanding of a text, as shown by Tündik et al. (2018). Rightly placed punctuation not only makes the text more readable and intelligible but can change the meaning of sentences, as well. Translated texts pose another challenge: Different languages expose different sentence structuring conventions and hence use punctuation very differently.

However, systems for automatic transcription of speech nowadays focus on minimizing the Word Error Rate (WER), which omits punctuation (He et al., 2011). As a result, the state-of-the-art systems are focused on the correct transcription of words and not necessarily correct segmentation of text or correct punctuation (Tündik et al., 2018).

*Equal contribution. Order determined by coin flip.

Therefore, attempts in improving the quality of such texts must also focus on a more precise prediction of punctuation. Consequently, this is an ongoing research effort in the NLP community. Recent developments in NLP (such as Transformers) offer new possibilities to tackle punctuation prediction effectively. Some of these attempts are discussed in Section 2.

Following recent attempts, we propose an ensemble system based on the Transformer architecture, where multiple models predict the punctuation symbols of a given text. The results are then combined and the final predictions are made. Our language-specific systems are able to predict punctuation for English, German, French, and Italian texts and are on par – if not better – with current state-of-the-art models that participated in the shared task.

Our main contributions include

1. the exploration of different Transformer-based models and identification of the most important features which affect the performance for this task, and
2. a showcase that the ensembling of differently trained models enhances the performance for the punctuation prediction task.

2 Related Work

Punctuation prediction tasks pose many challenges. One of them is the restricted input length thus restricted context for the Transformers. To solve the above mentioned limitation, Nguyen et al. (2019) used an overlapped chunk method (i.e., an overlapping sliding window) combined with a capitalization and a punctuation model to tackle the punctuation problem in long documents. First, the text is divided into chunks with overlapping segments. Second, a punctuation model (seq2seq LSTM, Transformer) predicts punctuation and capitalization for every segment. Lastly, overlapped

chunk merging combines chunks by discarding a defined number of tokens per overlapped chunk.

Courtland et al. (2020) changed the usual framework of punctuation prediction to predicting punctuation for the whole sequence rather than for single tokens. The authors used a feedforward neural network. Similar to Nguyen et al. (2019), they find that using a sliding window approach improves prediction performance. However, instead of producing multiple predictions for the same token, they sum activations before prediction and make inference afterwards.

Sunkara et al. (2020) used a joint learning objective for capitalization and punctuation prediction. The model input are sub-word embeddings. The authors used the pre-trained BERT model (BERT base truncated to the first six layers). They fine-tuned the model on medical domain data because the medical domain was in the main scope of this paper. They also fine-tuned the model for the punctuation prediction task. The authors used masked language learning objective while forcing half of the masked tokens to be punctuation marks.

Similarly, Nagy et al. (2021) also leveraged pre-trained BERT models (BERT base cased and uncased and a smaller version for English; multilingual and Hungarian-specific BERT versions for Hungarian). They added a two-layer multi-layer perceptron network with a soft-max output layer. The model also used a sliding window approach to enhance the results further. This model is trained to predict four labels: empty (no punctuation), comma, period and question mark.

Our approach differs from the above mentioned in using ensembling of multiple pre-trained Transformer-based models fine-tuned for the given task. Very importantly, our systems predict six different punctuation symbols for the punctuation prediction task.

Additionally, a multilingual Transformer was used as a part of our ensemble. We hypothesize that it would be able to capture more accurately the multilingual content of the EuroParl data. Furthermore, low-resource Latin languages might benefit from pre-training on more data, e.g., including other Latin languages.

In Section 3 we will discuss the datasets we used and the challenges they provide. The problem is described in Section 4 together with detailed description of our approach. Section 5 contains explanation of used hyperparameters, technical details and experimental setup. Section 6 presents our results and discusses the impact of used methods. Finally, a conclusion is drawn in Section 7.

Language	Training	Evaluation
English	11,028	10,521
German	11,495	10,207
French	12,276	13,366
Italian	10,379	10,502

Table 1: Mean token length per document in the training and evaluation dataset.

3 Dataset

The Europarl Parallel Corpus (Koehn, 2005) serves as the data source for the training, development, and test set. The surprise test set (of an undisclosed domain during evaluation) is an out-of-domain dataset that consists of a sample from the TED 2020 dataset (Reimers and Gurevych, 2020) with a low vocabulary overlap with the training data. As provided by the organizers of the shared task, samples in all datasets were lowercased and all punctuation marks were removed.

Subsequently, we outline challenges that we believe are especially relevant in solving this shared task and thus directly influenced our proposed system architecture.

3.1 Long Documents

As shown in Table 1, the mean token length is many orders of magnitudes longer than what typical Transformer architectures can process at once (typically up to 512 subtokens). It should be noted that some of the documents are especially long and can contain up to 100,000 tokens. The most obvious solution would be just to split documents into smaller sequences and subsequently merge predictions. However, this approach lowers the context with which a model is confronted and might lead to lower prediction quality (presumably at the beginning and end of a sequence).

3.2 Multilingual Content

To some extent, documents in the EuroParl Corpus contain multilingual content. As shown in the examples in Table 2, many documents contain names of people and areas that reflect the multilingualism of the participants of the European Parliament, i.e., members come from all over Europe. Therefore, using pre-trained models trained on monolingual

Sentence Excerpt
i agree completely with mr pöttering and with you too mr swoboda
the president of the european commission josé manuel durão barroso however

Table 2: Excerpts from the English training data that contain multilingual content.

data only may result in an inaccurate representation of this content.

3.3 Imbalanced Class Distribution

The class distribution of the training and evaluation set, as shown by Table 3, presents a rather typical situation in machine learning: Some of the classes have very few examples compared to the biggest classes. Neglecting this circumstance will likely lead to low performance for minority classes. Using typical techniques such as class-specific loss weights or data augmentation might improve performance to some extent. We have tried to reduce this problem by adding a model with altered loss weights to the ensemble.

Punctuation	Training	Development
:	43,133	9,490
?	44,290	9,815
-	80,916	18,335
.	1,396,166	319,751
,	1,759,686	401,095
0	30,454,904	6,985,003

Table 3: The distribution of class labels for English for the training and development set. 0 indicates the absence of a punctuation mark. The distributions for German, French and Italian are similar.

4 Methods

4.1 Problem Modelling

We modelled this problem as a token classification task. More precisely, each token is assigned a label representing the following punctuation symbol (if any). We concentrated our main efforts and focus on the full punctuation prediction. As such, we built all of the models to be able to predict all punctuation symbols. For the end of sentence prediction task, we mapped predictions of '.' '?' to 1 and the rest to 0.

Language	Transformers - Base
English	Electra, Longformer, MPNet, XLNet
German	BERT, Electra‡, XLM-RoBERTa
French	CamemBERT‡, Electra, XLM-RoBERTa
Italian	BERT, Electra‡, XLM-RoBERTa

Table 4: Transformer models that were used for each language-specific model. Models marked with ‡ were used twice: Once trained without weighted loss and once with weighted loss.

4.2 Transformers

The corner-stones of our systems are pre-trained Transformer models. We trained four different fine-tuned models for each language and combined the predictions using majority vote ensembling (see Section 4.5). Table 4 provides an overview.

Electra (Clark et al., 2020) is trained as a discriminator, and the authors suggest that it is more suitable for downstream sequence labelling tasks. In fact, we can further support this claim because this model architecture was the best-performing single model for all languages except French (see Table 5 and 6).

Both MPNet (Song et al., 2020) and XLNet (Yang et al., 2019) are trained (slightly differently) through permuted language modelling, allowing a better understanding of bidirectional contexts, which is often needed with punctuation. Both of these single models performed exceptionally well in our experiments.

Longformer (Beltagy et al., 2020), due to its local windowed attention with a task motivated global attention, can process larger sequence lengths (up to 4096) and perform well on the longer documents of this task.

XLM-RoBERTa (Conneau et al., 2019) is a multilingual transformer that is trained on over 100 languages. In our experiments it was demonstrated to be the best performing multilingual model.

The authors of CamemBERT (Martin et al., 2019) show that it performs exceedingly well in NER token classification. Moreover, the good performance translated to our French full-punctuation prediction experiments.

BERT (Devlin et al., 2018) has models pre-trained in multiple languages. We used language-

specific BERT models as part of German, French and Italian ensembles.

4.3 Sliding Window

As discussed earlier, documents in the corpus can be rather long, and typical Transformers cannot process such documents at once. Therefore, instead of simply splitting the documents into smaller segments, sequences are overlapped for inference. In other terms, a sliding window is applied, as suggested by Nguyen et al. (2019). Subsequently, the overlapped sequences are merged back together by discarding half of the overlapped tokens at the begging and end of each sequence. Our experiments have shown that an overlap of 40 tokens performs best. Consequently, we chose this overlap length for the final models.

4.4 Weighted Loss

For the German, French and Italian ensembles, we retrained the best performing model with weighted loss. We set the weights to three for the two least performing classes ('-', ':') and left them unchanged for the other classes (i.e., a weight of one). The idea is to increase recall for these classes by sacrificing overall performance, which, in return, helps an ensemble to create more accurate predictions.

Initially, we used inverted class frequencies as loss weights. However, this approach turned out to be too aggressive (worse minority class and overall performance). Further, we experimented with increasing minority class ('-', ':') weights. Initial experiments showed showed that weights set to three for minority classes and one for majority classes performed best on the development set. Our approach is rather heuristic, and further experimentation may lead to better results.

4.5 Majority Vote Ensembling

We did preliminary experiments in separate stacking models as mentioned in Wolpert (1992) as well as ensembling using the arithmetic average of class probabilities of single models as described in Goodfellow et al. (2014). However, one technique was shown to be more effective: majority vote ensembling. More concretely, all the models predict (i.e., vote) and the most voted label is then used as the final prediction. In case of a tie, the least common

Language	**Models**				**Ensemble**
English	Electra **0.940**	Longformer 0.934	MPNet **0.940**	XLNet 0.937	**0.943**
German	Electra **0.954**	XLM-RoBERTa 0.952	BERT 0.950	Electra‡ 0.953	**0.955**
French	Electra 0.923	XLM-RoBERTa 0.926	CamemBERT **0.930**	CamemBERT‡ 0.928	**0.933**
Italian	Electra **0.922**	XLM-RoBERTa 0.918	BERT 0.918	Electra‡ 0.919	**0.926**

Table 5: Positive class (sentence end) F1 results on the development set for all single models and the corresponding ensemble for sentence end prediction. Models marked with ‡ denote a model trained with weighted loss as described in subsection 4.4.

Language	**Models**				**Ensemble**
English	Electra **0.769**	Longformer 0.760	MPNet **0.768**	XLNet 0.763	**0.777**
German	Electra 0.803	XLM-RoBERTa 0.795	BERT 0.792	Electra‡ **0.805**	**0.812**
French	Electra 0.758	XLM-RoBERTa 0.761	CamemBERT 0.769	CamemBERT‡ **0.770**	**0.778**
Italian	Electra **0.746**	XLM-RoBERTa 0.732	BERT 0.741	Electra‡ 0.739	**0.755**

Table 6: Macro F1 results on the development set for all single models and the corresponding ensemble for full-punctuation prediction. Models marked with ‡ denote a model trained with weighted loss as described in subsection 4.4.

Language	Development			Test			Surprise Test		
	P	R	F1	P	R	F1	P	R	F1
English	0.93	0.96	0.94	0.93	0.95	0.94	0.84	0.75	0.80
German	0.95	0.96	0.96	0.95	0.96	0.96	0.89	0.77	0.82
French	0.92	0.94	0.93	0.92	0.94	0.93	0.82	0.72	0.77
Italian	0.91	0.95	0.93	0.90	0.95	0.93	0.83	0.71	0.77

Table 7: Ensembling positive class (sentence end) F1 results on the development, test and surprise test set for sentence end prediction.

Language	Development			Test			Surprise Test		
	P	R	F1	P	R	F1	P	R	F1
English	0.82	0.75	0.78	0.81	0.75	0.77	0.65	0.59	0.62
German	0.82	0.80	0.81	0.82	0.80	0.81	0.66	0.65	0.65
French	0.80	0.76	0.78	0.78	0.77	0.77	0.63	0.60	0.61
Italian	0.77	0.74	0.76	0.77	0.74	0.75	0.57	0.55	0.56

Table 8: Ensembling Macro F1 results on the evaluation, test and surprise test set for punctuation prediction.

label is chosen. Additionally, predictions for a hyphen are counted twice – mainly to increase the performance for the worst-performing label (which was the case for all languages). Our experiments on the development set have shown that this leads to an increase of 1-2% Macro F1 score for all languages compared to the single best performing model.

5 System Architecture

5.1 Hyperparameter Setup

At the beginning of development, we empirically determined what characteristics of the model and fine-tuning correlate with better performance. For fine-tuning, five epochs performed consistently well for all transformer architectures. Due to the large document size, the larger the maximum sequence length, the better the performance. To our surprise, there were no significant differences between the performance of cased vs. uncased Transformers on our lower-cased data.

5.2 Technical Implementation

For the training of our models, we used the Simple Transformers [1] library, a wrapper for the Hugging Face [2] library, that allows for fast experimenting. As the Simple Transformers library does not support weighted loss training, we have adapted the relevant code for this purpose.

[1] https://simpletransformers.ai
[2] https://huggingface.co

5.3 Experimental Setup

We trained all of the models on a single T4 GPU instance. Our final models shared some of the hyperparameters, namely a learning rate of $4e{-}5$, a batch size of 16 (four for Longformer) and the maximum sequence length (512, 4096 for Longformer). We trained each model for five epochs.

6 Results & Discussion

Our results for sentence end prediction and full punctuation prediction can be seen in Table 7 and Table 8, respectively. They demonstrate the high capability of using Transformers in predicting punctuation marks. Especially for sentence end prediction, the F1 scores are well above 90% for all languages. We hypothesize that it is because usage of sentence end punctuation is less ambiguous – it is consistently and grammatically correctly used in the data. For full punctuation prediction, the overall performance is significantly lower for all languages. The full punctuation prediction task is more difficult not only because of the existence of more labels, but also because some of the labels might not follow strict grammatical rules. For example '-' or a ':' can be used differently due to different styles of linguistic expressions, while a label such as a comma might be misplaced due to human error.

With respect to our system, sliding windows are a simple way to improve performance when an input sequence is much longer than what a model can actually process. However, this performance gain is limited, and as of now, it is not clear how

this compares to a model that can process much longer sequences. Observing results we have obtained from single models at Table 5 and 6 for both subtasks we can see that the model architecture has an effect on performance. Within our experiments, majority vote ensembling further enhances performance.

7 Conclusion

In this paper, we showed that the ensembling of diversely trained Transformers can yield significant improvement and allows for good generalisation for punctuation prediction in out-of-domain examples. From this work, it can be seen that combining different Transformers can be really beneficial. However, further work is needed to determine if more advanced ensembling techniques could further increase the quality of the predictions.

Acknowledgments

We want to thank Simon Clematide and Phillip Ströbel for their valuable inputs and the Departement of Computational Linguistics for providing us with the necessary technical infrastructure.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. Efficient automatic punctuation restoration using bidirectional transformers with robust inference. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 272–279.

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*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Xiaodong He, Li Deng, and Alex Acero. 2011. Why word error rate is not a good metric for speech recognizer training for the speech translation task? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5632–5635. IEEE.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Attila Nagy, Bence Bial, and Judit Ács. 2021. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019. 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)*, pages 1–5. IEEE.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Monica Sunkara, Srikanth Ronanki, Kalpit Dixit, Sravan Bodapati, and Katrin Kirchhoff. 2020. Robust prediction of punctuation and truecasing for medical asr. In *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, pages 53–62.

Máté Akos Tündik, György Szaszák, Gábor Gosztolya, and András Beke. 2018. User-centric evaluation of automatic punctuation in asr closed captioning.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.