University of Regensburg @ SwissText 2021 SEPP-NLG: Adding Sentence Structure to Unpunctuated Text

Gregor Donabauer
University of Regensburg
Regensburg, Germany
gregor.donabauer@stud.uni-regensburg.de

Udo Kruschwitz
University of Regensburg
Regensburg, Germany
udo.kruschwitz@ur.de

Abstract

This paper describes our approach (URmSBD) to address the shared task on *Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG)* organised as part of *SwissText 2021*. We participated in *Subtask 1 (fully unpunctuated sentences – full stop detection)* and submitted a run for every featured language (English, German, French, Italian). Our submissions are based on pre-trained BERT models that have been fine-tuned to the task at hand. We had recently demonstrated, that such an approach achieves state-of-the-art performance when identifying end-of-sentence markers on automatically transcribed texts. The difference to that work is that here we use language-specific BERT models for each featured language. By framing the problem as a binary tagging task using the outlined architecture we are able to achieve competitive results on the official test set across all languages, with *Recall, Precision, F1* ranging between 0.91 and 0.96 which makes us joint winners for *Recall* in two of the languages. The official baselines are beaten by large margins.

1 Introduction

Text normalization has always been a core building block of natural language processing aimed at converting some raw text into a more convenient, standard form (Jurafsky and Martin, 2020). Besides tokenization, stemming and lemmatization this process includes sentence segmentation. What is interesting though is that text pre-processing and normalization is by no means a solved challenge.

The *SwissText 2021 Shared Task 2: Sentence End and Punctuation Prediction in NLG Text* is concerned exactly with this problem area. The goal is to develop approaches for sentence boundary detection (SBD) in unpunctuated text. Providing suitable solutions means fostering readability and restoring the text's original meaning.

We took part in *Subtask 1 (fully unpunctuated sentences – full stop detection)* of this challenge and did so for all featured languages. This report starts by contextualising the task as part of a short discussion of related work. We will then introduce our methodology, briefly describe the data and report results. Finally we present some discussion and conclusions.

2 Related Work

Sentences are considered as a fundamental information unit of written text (Jurafsky and Martin, 2020; Levinson, 1985). Therefore, many NLP pipelines in practice split text into sentences. *Fact checking* is just one – currently very popular – challenge where the automated detection of sentences within a stream of input data is essential. Fact checkers are increasingly turning to technology to help, including NLP (Arnold, 2020). These tools can help identify claims worth checking, find repeats of claims that have already been checked or even assist in the verification process directly (Nakov et al., 2021). Most such tools rely on text as input and require the text to be split into sentences (Donabauer et al., 2021). For this and other application areas sentence segmentation will remain a challenging task despite the fact that recent developments suggest that for some NLP tasks it is possible to achieve state-of-the-art performance without conducting any pre-processing of the raw data, e.g. (Shaham and Levy, 2021).

Sentence Boundary Detection (SBD) is an important and actually well-studied text processing step but it typically relies on the presence of punctuation within the input text (Jurafsky and Martin, 2020). Even with such punctuation it can be a difficult task, e.g. (Gillick, 2009; Sanchez, 2019), and traditional approaches use a variety of architectures including CRFs (Liu et al., 2005) and combinations

of HMMs, maximum likelihood as well as maximum entropy approaches (Liu et al., 2004). With unpunctuated texts (and lack of word-casing information) it becomes a lot harder as even humans find it difficult to determine sentence boundaries in this case (Stevenson and Gaizauskas, 2000). Song et al. (2019) simplify the problem by aiming to detect the sentence boundary within a 5-word chunk – using YouTube subtitle data. Using LSTMs they predict the position of the sample's sentence boundaries but did not consider any chunks without sentence boundary. Le (2020) presents a hybrid model (using BiLSTMs and CRFs) originally used for NER that was evaluated on SBD in the context of conversational data by preprocessing the CornellMovie-Dialogue and the DailyDialog datasets to obtain samples that neither contain sentence boundary punctuation nor word-casing information (they also predict whether the sentence is a statement or a question). Du et al. (Du et al., 2019) present a transformer-based approach to the problem, but they assume partially punctuated text and word-casing information. Recently, it was shown that a simple fine-tuned BERT model was able to improve on the state of the art on fully unpunctuated case-folded input data (Donabauer et al., 2021).

3 System

3.1 General Architecture of UR-mSBD

The system architecture we use is adopted from our previous work that achieved state-of-the-art performance on a very similar task (Donabauer et al., 2021). That architecture demonstrated the suitability of a BERT-based token classification approach for sentence end prediction in the context of improving text processing pipelines for fact-checking. The underlying idea is to treat the restoration of sentence boundary information as a problem similar to IO-tagging in named entity recognition. For the implementation we refer to our GitHub repository[1]. The last token of every sentence, marking the occurrence of a sentence boundary punctuation mark to follow up, is labeled with EOS. In our previous work we predicted the beginning of a sentence rather than its end. We therefore labeled the first token of every sentence with BOS. Out-of-context labels O are assigned to all other tokens of the text.

We fine-tuned a pre-trained BERT model on the problem and obtained high F1 scores for the desired positive class (sentence end) outperforming alternative approaches on different datasets. We use a softmax classification head predicting the label (EOS or O) by the highest probability at each token.

[1] https://github.com/doGregor/SBD-SCD-pipeline

3.2 Adjustments for the Shared Task

We apply two changes to the model fine-tuning process for this shared task as follows:

- First of all, we are faced with four different languages and not just English texts. The two obvious options would be to use a multilingual language model or to choose a different language-specific pre-trained model for each of the languages, i.e. German, French, English and Italian. We decided to adopt language-specific BERT-base models as Nozza et al. (2020) reports that this yields better results than using mBERT, pre-trained on a multilingual corpus.

- Secondly, we change the process of sample construction. We handle the unpunctuated input text as one long chain of words. We originally split this chain in samples of 64 words and fine-tuned the model with a maximum sequence length of 128 BERT-specific tokens. Further experiments have shown that utilizing token sequences as long as possible (512 BERT tokens) yields the best results. Therefore, we pre-process the raw text data by sending it through the model's tokenizer first. Each time a batch of iterated words fits 512 BERT tokens we create a sample from it. If a word at the transition between two samples would be ripped apart (as adding it entirely to the current sample would exceed 512 tokens), we put it at the beginning of a new sample and pad the rest of the previous one with special PAD tokens.

All other hyperparameters are kept in line with Donabauer et al. (2021), namely using an epoch number of 3 and a batch size of 8 per device. Since we run it on 3 GPUs simultaneously the batch size per iteration increases to 24. We also evaluated our approach on the datasets with tuned hyperparameters. However, it turned out that increasing the number of epochs to 5 leads to a deterioration of results.

4 Data and Setup

We participated in *Subtask 1 (fully unpunctuated sentences – full stop detection)* of SwissText's SEPP-NLG Shared Task 2.

Before addressing the experimental setup we briefly describe the provided data sets. The challenge's domain are NLG texts. Since there are no corpora that feature such data nor manually corrected versions the organizers selected Europarl[2] as source. This corpus includes transcribed text data originating from spoken text in many different languages. The data come along in lowercase format and are already split up into tokens. Sentence boundary punctuation is removed. Instead labels are assigned that mark upcoming sentence ends. The last token of each sentence is labeled with '1', all remaining with '0'.

The data are provided as multiple tab-separated value files grouped by each language and set. The number of tokens per language and dataset is reported in Table 1. We explain our pre-processing with respect to a single set for a single language, e.g. the English evaluation set. Firstly, we read each tsv file one after the other and concatenate all tokens and labels as two long lists. During reading we save the order and length of the input files. By that we are able to reconstruct the original structure of the files later on. The list of tokens is fed into the model-specific tokenizer. If tokens are not recognized properly we replace them with 'nan'. Each time a batch of 512 BERT tokens is filled, we create a sample from it. Data are saved in CoNLL-2003 format (Tjong Kim Sang and De Meulder, 2003). Tokens and labels are separated horizontally with spaces. Samples are separated vertically with empty lines. We use the tokenizer during pre-processing only to calculate the number of BERT tokens at each input word. The samples themselves consist of plain text tokens. Thus dimension and order of predicted labels correspond to the structure of the processed tsv files. We can then simply map our output to the words in the input data.

As mentioned earlier, we make use of language specific models rather than mBERT. We briefly describe the respective models and the corpora they are trained on.

- *English:* Classic BERT base uncased model, trained on English lowercase text (Devlin et al., 2019).
- *German:* BERT base uncased model, trained on 16GB monolingual German corpus by dbmdz (MDZ Digital Library team at the Bavarian State Library)[3].
- *French:* BERT base uncased model, trained on 71GB monolingual French corpus (Le et al., 2020).
- *Italian:* BERT base uncased model, trained on 81GB monolingual Italian text by dbmdz.

We make use of the PyTorch[4] version of the Python huggingface[5] transformers library to access models and run fine-tuning. We execute the scripts on 3 Nvidia GeForce RTX 2080 Ti GPUs with an overall memory size of 33GB.

5 Results

5.1 Baselines

The official baseline is produced using the spaCy NLP package. The organisers report scores for different pipeline versions and we are describing the best performing one for every language in Table 2. The official evaluation metrics are Precision, Recall and F1-score of the positive class label (i.e., sentence end).

As Table 2 illustrates, F1-scores for English, German and French are ranging from 0.32 to 0.47. For Italian the F1-metric is collapsing to 0.01, caused by a very low Recall of 0.00.

5.2 UR-mSBD

We summarise the results obtained when running our system, UR-mSBD, on the test data. For each language we also include scores obtained on the dev set as well as the *surprise test set* that was introduced to check the generalizability of the different approaches.

Table 3 presents the results for the English data, Table 4 for German, Table 5 for French, and Table 6 presents the results for the Italian test data.

We see overall consistently high scores for all three metrics and across all languages when looking at the official test sets. An average of F1=0.93 aggregated over all languages places us just one percentage point behind the top performance. Looking at Recall, we actually end up being joint winners for the German and French test data.

[2] https://opus.nlpl.eu/Europarl.php

[3] https://github.com/dbmdz/berts

[4] https://pytorch.org/

[5] https://huggingface.co/

Language	Train	Dev	Test	Surprise Test
English	33,779,095	7,743,489	10,039,222	1,081,910
German	28,645,112	6,358,683	9,575,861	979,982
French	32,690,367	8,781,593	11,297,534	1,143,911
Italian	28,167,993	7,194,189	10,193,542	985,448

Table 1: Number of tokens in the respective data sets for each language.

Dataset	Precision	Recall	F1
Dev EN	0.49	0.23	0.32
Test EN	0.49	0.24	0.32
Dev DE	0.51	0.44	0.47
Test DE	0.49	0.44	0.46
Dev FR	0.71	0.24	0.36
Test FR	0.63	0.24	0.35
Dev IT	0.64	0.00	0.01
Test IT	0.51	0.00	0.01

Table 2: Highest baseline scores for EN, DE, FR, IT.

Dataset	Precision	Recall	F1
Dev	0.92	0.92	0.92
Test	0.91	0.92	0.92
Surprise Test	0.82	0.68	0.74

Table 3: UR-mSBD scores for English.

Dataset	Precision	Recall	F1
Dev	0.96	0.95	0.95
Test	0.94	0.96	0.95
Surprise Test	0.89	0.73	0.80

Table 4: UR-mSBD scores for German.

Dataset	Precision	Recall	F1
Dev	0.94	0.93	0.93
Test	0.93	0.94	0.93
Surprise Test	0.83	0.70	0.76

Table 5: UR-mSBD scores for French.

Dataset	Precision	Recall	F1
Dev	0.93	0.91	0.92
Test	0.91	0.93	0.92
Surprise Test	0.84	0.67	0.74

Table 6: UR-mSBD scores for Italian.

The highest scores are reported for German (with Precision at 0.94, Recall at 0.96 and F1 at 0.95). All the scores for the test sets are above 0.90. For the surprise test set the results drop quite a bit but are still reasonably high given the data is not representative of the data the system was trained on.

Across the board all the baselines were beaten by large margins.

6 Discussion

For all featured languages our fine-tuned BERT-based predictions are performing very well with results for all three metrics (P/R/F1) in the 90s and being very competitive when looking at the other submissions for this shared task. This first of all demonstrates the power of transformer-based models and confirms findings we reported previously (Donabauer et al., 2021).

The fact that the baselines were outperformed by such large margins is perhaps a sign that non-neural approaches are not competitive for the task and data at hand.

We note that our approach performed best for German texts which might be caused by high similarity between the data the model was pre-trained on and the data sampled to be training, dev and test sets for this task. It will be worth exploring whether for different data samples we observe a similar pattern or whether the differences are in fact not significant.

Taking a slightly broader perspective, we observe that the scores obtained here are similar to what we obtained when running our sentence boundary detection algorithm on a dataset comprising transcribed lectures given at Stanford University first proposed by Song et al. (2019), and the DailyDialog dataset (Li et al., 2017) but that extending these datasets or creating a hybrid version resulted in significant drops in performance (Donabauer et al., 2021). It would therefore be interesting to see whether other approaches illustrate similar patterns.

Another general pattern we read into the results is that there are only small differences when comparing results on the dev sets with the results on the test sets. We conclude that our approach can generalize to unseen data as long as the training data is representative for the data used for testing.

The approach does however generalise less well to out-of-domain ('surprise') data with F1-scores dropping between 0.15 and 0.18, compared to the Europarl sets. We still consider the results to be reasonably well though given they are on average over all languages only 0.03 behind the top-performing system.

7 Conclusions

We framed the task of full-stop prediction (*Subtask 1 of Shared Task 2 at SwissText 2021*) as a binary classification task over all input tokens identifying whether each of these tokens should indicate the position of a full stop or not. Fine-tuning language specific pre-trained BERT models for each of the four tasks resulted in competitive results. Given the small difference in F1 of 0.01 compared to the top results reported for this competition for three of the languages (as well as aggregated over all languages) we will await statistical significance tests as our results may well turn out to be on par with the top results in this task.

Acknowledgements

This work was supported by the project *COURAGE: A Social Media Companion Safeguarding and Educating Students* funded by the Volkswagen Foundation, grant number 95564.

References

Phoebe Arnold. 2020. The challenges of online fact checking. Technical report, Full Fact, London, UK.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gregor Donabauer, Udo Kruschwitz, and David Corney. 2021. Making sense of subtitles: Sentence boundary detection and speaker change detection in unpunctuated texts. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion)*, New York, NY. ACM.

Jinhua Du, Yan Huang, and Karo Moilanen. 2019. AIG Investments.AI at the FinSBD task: Sentence boundary detection through sequence labelling and BERT fine-tuning. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 81–87, Macao, China. Association for Computational Linguistics.

Dan Gillick. 2009. Sentence boundary detection and the problem with the u.s. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, page 241–244, USA. Association for Computational Linguistics.

Daniel Jurafsky and James Martin. 2020. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Current draft of third edition (30 Dec 2020).

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

The Anh Le. 2020. Sequence labeling approach to the task of sentence boundary detection. In *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, ICMLSC 2020, page 144–148, New York, NY, USA. ACM.

Joan Persily Levinson. 1985. *Punctuation and the orthographic sentence: a linguistic analysis*. Doctoral dissertation, City University of New York.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2004. Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 64–71, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 451–458, USA. Association for Computational Linguistics.

Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño,

Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. *CoRR*, abs/2103.07769.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific BERT models. *CoRR*, abs/2003.02912.

George Sanchez. 2019. Sentence boundary detection in legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 31–38, Minneapolis, Minnesota. Association for Computational Linguistics.

Uri Shaham and Omer Levy. 2021. Neural machine translation without embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 181–186, Online. Association for Computational Linguistics.

Hye Jeong Song, Hong Ki Kim, Jong Dae Kim, Chan Young Park, and Yu Seop Kim. 2019. Inter-sentence segmentation of YouTube subtitles using Long-Short Term Memory (LSTM). *Applied Sciences (Switzerland)*, 9(7).

Mark Stevenson and Robert Gaizauskas. 2000. Experiments on sentence boundary detection. In *Proceedings of the sixth conference on Applied natural language processing* -, pages 84–89, Morristown, NJ, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* -, volume 4, pages 142–147, Morristown, NJ, USA. Association for Computational Linguistics.