Comprehensive Punctuation Restoration for English and Polish

Michał Pogoda
Wrocław University of Science and Technology, Poland
michal.pogoda@pwr.edu.pl

Tomasz Walkowiak
Wrocław University of Science and Technology, Poland
tomasz.walkowiak@pwr.edu.pl

Abstract

Punctuation restoration is a fundamental requirement for the readability of text derived from Automatic Speech Recognition (ASR) systems. Most contemporary solutions are limited to predicting only a few of the most frequently occurring marks, such as periods, commas, and question marks — and only one per word. However, in written language, we deal with a much larger number of punctuation characters (such as parentheses, hyphens, etc.), and their combinations (like parenthesis followed by a dot). Such comprehensive punctuation cannot always be unambiguously reduced to a basic set of the most frequently occurring marks. In this work, we evaluate several methods in the comprehensive punctuation reconstruction task. We conduct experiments on parallel corpora of two different languages, English and Polish — languages with a relatively simple and complex morphology, respectively. We also investigate the influence of building a model on comprehensive punctuation on the quality of the basic punctuation restoration task.

1 Introduction

The task of restoring punctuation can be crucial for the readability of text derived from ASR systems. As Tündik et al. (2018) has shown, a lack of punctuation in transcription can have a greater negative impact on readability than a large number of word transcription errors. In recent years, punctuation prediction was most often approached as a token classification task (Tilk and Alumäe (2016), Kim (2019), Alam et al. (2020)). In this context, the target labels are often reduced to only a few most frequently occurring marks, such as periods, commas, and question marks. However, in written language, we deal with a much larger number of characters (such as parentheses, hyphens, etc.). The usual approach is to try to reduce those punctuation marks into the basic set via role similarity (e.g., semicolon and exclamation marks are often reduced to periods) or discard them entirely (Tilk and Alumäe (2016), Żelasko et al. (2018)). However, such a process always comes with a loss of information. Furthermore, a word can end with more than one punctuation mark — for example, the end of parenthesis can coincide with the end of a sentence, resulting in the combination of these marks into '). '. Predicting only a period in such a place would quite strongly violate the structure of the original statement (see Table 1). We propose a new approach to the punctuation restoration task — Comprehensive Punctuation Restoration — where the task will be to restore all the original punctuation in the text (i.e., without any reduction) in a token classification manner.

In the following work, we explore the possibility of generating a manageable-sized set of labels directly from the dataset, based on the percentile of punctuation cases present in the set. We measure increased recall by using broader class sets and a potential cost in terms of precision. In addition, we test whether models trained on more narrowly defined classes will suffer (or gain) on a reduced, conventionally defined 4-class task.

We conducted our research on a parallel corpus of Polish and English — two languages with very different levels of morphological complexity (Łockiewicz and Jaskulska, 2017). With this approach, we can directly compare a set of semantically identical and volumetrically very similar datasets and see how well our results generalize. We would be able to catch if a trend in our results was very specific to a single language.

In summary, in this paper we made the following contributions:

- We propose an approach to generate a comprehensive punctuation label set directly from the dataset rather than some predefined marks.
- We evaluate how increasing the size of gen-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4610–4619
November 7–11, 2021.

Type of restoration	Text
4 classes	our way is to honor every religion and every nation according to their paths, as it is written in the book of prophets because every nation will go in the name of its lord.
4 classes + mapping	our way is to honor every religion and every nation according to their paths, as it is written in the book of prophets, because every nation will go in the name of its lord.
full restoration	our way is to honor every religion and every nation according to their paths, as it is written in the book of prophets: 'because every nation will go in the name of its lord.'

Table 1: Comparison of the same quote with different approaches to punctuation reduction. Depending on what characters the system can restore, you can get punctuation capable of representing different structures of an expression. Converting to a set of basic characters (in this case a colon to a comma), while helpful, cannot always preserve the entire meaning of the original punctuation.

erated label set affects the ability to restore complete punctuation.

- We investigate whether using a large, narrowly-defined set of labels affects the performance of the model on a frequently used, basic set of 4 classes: PERIOD, COMMA, QUESTION, and OTHER.

Also, to the best of our knowledge, our proposal is the first publicly described research for restoring punctuation for the Polish language. [1]

2 Related Work

The first approach to punctuation restoration (in the sense of restoring punctuation marks) has been proposed by Beeferman et al. (1998). They introduced a model based on the Markov chain, designed for restoring commas in the output of ASR systems. In the field of deep learning, the punctuation restoration task is often approached with bidirectional recurrent neural networks. Most often LSTM and GRU architectures are used. Although LSTM networks are often — computational performance aside — considered better than GRUs in the general case (Yang et al. (2020), Weiss et al. (2018)), it is reported in several papers that GRUs outperformed LSTMs in the punctuation restoration task (Tilk and Alumäe (2016), Hládek et al. (2019)).

In Tilk and Alumäe (2016) authors explored the possibility of using bidirectional recurrent networks with attention for the punctuation restoration on the Estonian language. They provided their code[2] with the publication and we will be using it in our research as an example of a recurrent network model.

Lately, an interesting approach based on LSTMs was proposed by Li and Lin (2020), where they tried to create a single model for restoring punctuation for 43 languages using language-independent BPE tokenization. They also included Polish in the training set, however, authors did not publish per-language results.

In recent years, large, pre-trained models based on transformer architecture (Vaswani et al., 2017) seem to perform best on a number of NLP tasks, including punctuation restoration. Perhaps the most comprehensive comparison of various transformer encoder models in the task of punctuation restoration is done by Alam et al. (2020), where the authors compared a number of models based on different variants of pre-trained BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and AlBERT (Lan et al., 2020) encoders as a base of their model. They've shown that generally larger pre-trained models were better than the smaller ones in punctuation restoration and that between models of the same size, generally, RoBERTa was better than both AlBERT and BERT. They also showed that XLM (cross-lingual models) variants of RoBERTa were slightly worse than English-only ones. The authors of the paper published their code and we also used it in our research.

In Yi et al. (2020) authors show that punctuation restoration can also benefit from multitask learning (POS tagging being the secondary task in their

[1] There is the Polish language mentioned as a part of a multilingual model in Li and Lin (2020), however, the authors did not publish per-language results.

[2] https://github.com/ottokart/punctuator2

work). They trained a single BERT-based model with 2 token classification heads — one for Punctuation restoration and one for POS tagging. While the weights in the heads were separate in the task, the BERT core was shared. They have shown that such a form of regularization can help in the punctuation restoration of unseen data.

Our take on the punctuation restoration task is inspired by the work of Omelianchuk et al. (2020), where the authors used an approach with automatic generation of a set of labels from the data to approximate the capabilities of sequence to sequence models token classification. They did it in the context of the grammatical error correction task.

3 Dataset

Lang	Total	Train	Dev	Test
En	15.27M	12.23M	1.53M	1.52M
Pl	12.69M	10.15M	1.27M	1.27M

Table 2: Number of words in the data set with decomposition into training, test and validation sets.

To be able to research comparable corpora for different languages, similarly to Vandeghinste et al. (2018), we used a parallel corpus from the Europarl v7 dataset[3]. The corpus is extracted from the proceedings of the European Parliament and translated into multiple languages. Specifically, we use the parallel corpus of Polish and English taken from proceedings from 01/2007 to 11/2011. The corpus is made up of 15.27M words (English) and 12.82M words (polish) divided into sentences, with each sentence on a separate line. As some of the lines are very short and contain e.g. only a single number, we removed all of the lines that had fewer than 4 words as a preprocessing step. Then we divided the corpus randomly into training, validation, and test collection in the ratio 8/1/1 (line-wise). See Table 2 for information on the size of each collection.

The text preprocessing step consisted only of normalization of all whitespace characters (including newline) into a single space. The decision was motivated by the fact that whitespace is mostly connected with formatting rather than punctuation. In the specific case of the dataset we used, new lines were used to separate sentences. However, if the dataset was annotated in a way that whitespace formatting was meaningful (ie., using newlines or tabulations for paragraph splitting), this step could be skipped and attempts could be made to also reproduce subtle differences in whitespace.

[3]https://www.statmt.org/europarl/

Lang	Percentile	Classes	Lowest support
En	90th	7 + 1	22,767
	95th	12 + 1	14,115
	99th	23 + 1	2,007
	100th	513 + 1	1
Pl	90th	6 + 1	21,678
	95th	11 + 1	14,332
	99th	28 + 1	1,308
	100th	716 + 1	1

Table 3: The number of punctuation classes based on the percentile of punctuation retained. The +1 stands for the additional class representing 'no punctuation' — single whitespace only.

After preprocessing, the text was broken into tokens based on the occurrence of any non-alphanumeric character (including whitespace). Each alphanumeric sequence was considered a single token, and each non-alphanumeric sequence following it was considered a label of that token. The set of all unique non-alphanumeric sequences was considered the largest possible set of punctuation labels for this specific dataset.

The classes from the label set were then sorted by their frequency of occurrence in the text. Obviously, in most cases, by far the most represented class was single whitespace (that made up 88.36% of all labels in the English dataset and 85.14% in Polish). Overall, we got 513 unique classes for the English version and 716 classes for Polish one. In both cases, there was a long tail of underrepresented classes. Such classes consisted mainly of combinations of rare marks (eg. “”=.”) or very long strings of punctuation characters (e.g., “ [.../...] [”). In the case of the English dataset, there were 330 classes with fewer than 5 occurrences and 186 classes with only one occurrence. In the case of Polish, such a long tail was even longer, with 486 classes with less than 5 occurrences and as many as 287 with only one occurrence.

As stated in the introduction, the goal of this work is to reproduce as much of the original punctuation as possible. Because of that, in the test set no class reduction was done and all the original labels were put there in unchanged form (even if the class had only one occurrence in the test set and no occurrence in the training set). However, training a model on classes that had only a few samples would be impractical. Because of that, the

training and validation sets were reduced in such a way that we maximize class coverage. To do that, we created a set of label subsets according to a minimum number of classes to achieve a given percentile (not counting single whitespace). The percentiles with a corresponding number of classes are presented in Table 3.

As the last step, we created another version of the label set with the number of classes reduced to a commonly used quadruple of labels: COMMA, PERIOD, QUESTION, OTHER. We used this version of the dataset to test whether training models on a larger, more narrowly defined set of labels would have negative effects relative to models trained on a small label set. To map the comprehensive label set into a simple label set we used inclusion criteria. I.e., if the character "." was present in the comprehensive label (e.g. "". "), it would be mapped into PERIOD class. If more than one base class were found in the comprehensive label, then the more frequently represented one would be chosen (the precedence order was a comma, period, question). If no base label was found, OTHER class was assigned. This variation of label set will be referred to as "Reduced"

4 Experimental Setup

4.1 Overview

For our experiments, the first architecture we used was bidirectional GRU with an attention model, described in Tilk and Alumäe (2016). Originally, the authors of this paper also tested their solution on a corpus derived from Europarl v7 (though not a parallel one). In their case, they used a total of 8 classes (consisting of 7 punctuation marks plus class representing no punctuation). This set of labels will be further marked as "$\text{Base}_{\text{Tilk}}$". As for hyperparameters, we used the ones suggested by the authors (learning rate of 0.02 and hidden layer size of 256).

The next set of architectures we examined were base-sized transformer models derived from Alam et al. (2020). We used BERT and RoBERTa for our study of the English dataset and Bert for Polish. In Alam et al. (2020), the authors provide a standard set of 4 classes — period, comma, question mark, and the 'other' class (which also contains no punctuation). This set of 4 classes will be marked as "$\text{Base}_{\text{Alam}}$". For the pre-trained Polish Bert models, we used the one trained by Kłeczek (2020), hosted on huggingface model repository[4]. We used the cased version because the author recommends using it over the uncased version. The only changes we made to the original code from Alam et al. (2020) are those allowing us to change the scope of the predicted classes and to incorporate more pre-trained models (Polish ones). For the hyperparameters, we used a learning rate of 10^{-5}, batch size of 8, augmentation rate of 0.15 with alpha-sub and alpha-del set to 0.4. We trained each model for 10 epochs.

We first trained the described models on the dataset with labels mapped to the original set of labels (i.e., the sets that were used in the original implementations and marked as "Base"). Base sets were mapped to comprehensive labels by matching the most frequently represented label containing a character from the base set. For example, the base label "!" would be mapped to the comprehensive label "! ". Labels that were not mapped were replaced with a single whitespace label (" "), representing no punctuation.

We then incrementally increased the number of labels in the training and validation sets such that they covered the 90th, 95th, and 99th percentiles of all the original punctuation (see Table 9 and 10). On those models, we examined how increasing the size of a training label set would affect the precision and recall of punctuation in the original texts, on the test set. In each experiment, the test set contained all original labels (i.e., 513 for the English set and 716 for the Polish set).

At last, we trained the models on the reduced dataset. Those models will be mainly used as a baseline to check whether training the models on comprehensive label sets would have a positive or negative effect on the quality of model performance for the core classes. It is worth noting that the models trained on this set will attempt to predict the labels greedily (i.e., the models are trained to predict the label ". " even when the label "". " was originally present). For this reason, these models will achieve lower average precision on the comprehensive punctuation restoration task.

All the experiments were performed on following hardware: RTX 2080 Ti, Intel(R) Xeon(R) CPU E5-2650, 503Gb of RAM. The longest single fine-tuning process took 5 hours 43 minutes (BERT on English dataset).

[4] https://huggingface.co/dkleczek/bert-base-polish-cased-v1

4.2 Metrics

4.2.1 Token classification metrics

For the comprehensive punctuation restoration task, we used precision (P), recall (R), and f1 computed as micro-averages of all classes excluding a single whitespace class (i.e., the dominant class corresponding to the absence of punctuation). Predictions are marked as correct only if the model predicted the exact class (i.e., predicting '. ' for a token with ground truth label '). ' would be counted as an error). For a task with reduced labels, we used precision, recall, and f1 for classes COMMA, PERIOD, and QUESTION. We also computed the macro average of those metrics under the TOTAL section.

4.2.2 mRS

Token classifications metrics are very strict (i.e., if the true test label was '). ' and the model predicted '. ' it would still count as a full error). Intuitively, if the model predicted a label that had some common part with a true label, it should be counted as a better score than predicting a completely wrong one. To address this issue, we used a third metric — mean Ruzicka similarity (mRS). Ruzicka similarity (Deza and Deza, 2009) is a weighted version of Jaccard similarity that allows us to work on a multiset (e.g. labels like '...'). It has values in the range (0,1) where 1 is achieved for a perfect match, higher values mean better results. In our application, this metric is defined as follows:

$$RS(\mathbf{P}, \mathbf{T}) = \frac{\sum_{k=1}^{c} min\left(p_k, t_k\right)}{\sum_{k=1}^{c} max\left(p_k, t_k\right)}$$

Where

$$\mathbf{P} = [p_1, p_2, ..., p_c]$$

$$\mathbf{T} = [t_1, t_2, ..., t_c]$$

are predicted and ground-truth labels of the same token, represented by a vector consisting of the count of all single-character punctuation marks in that label (excluding whitespace).

To compute mean RS we just average RS metric over all labels, skipping the tokens where the ground-truth label is whitespace only (i.e., no punctuation).

$$mRS = \frac{\sum_{i=1}^{N} RS(\mathbf{P}_i, \mathbf{T}_i)[\mathbf{T}_i \neq \mathbf{0}]}{\sum_{i=1}^{N}[\mathbf{T}_i \neq \mathbf{0}]}$$

where $\mathbf{P}_i$ is predicted label for i$^{\text{th}}$ token, $\mathbf{T}_i$ is ground-truth label for i$^{\text{th}}$ token and N is a total number of tokens.

5 Results for English

Example predictions (on an excerpt from the test set) from the best model for English (RoBERTa) trained on a different number of training labels is shown in Table 4, whereas the metrics for all experiments are presented in Table 5.

As expected, increasing the number of classes on which the model was trained increases the average recall (R). Depending on the method, the increase over the method's native class list was between 12 and 14 percentage points. As for averaged precision (P), its clear decrease was observed only in the case of the BiGRU model. In models based on pre-trained Berts, the highest precision was obtained with an increased number of classes. Since the fluctuation of precision with increasing label set was relatively small compared to the gain on recall, the f1 metric in each case increased with the increasing number of classes. Also, the less rigid mRS metric showed an average gain of about 14 points when using models trained on 99$^{\text{th}}$ percentile. This number shows how much we would be losing when we would reduce the punctuation to the base set.

Table 6, on the other hand, presents a comparison of the performance of the models on the reduced set of labels. It can be observed that the number of labels (L column) on which the model was trained did not have a major impact on the quality of the task in the basic formulation of the problem. The only clear decrease can be seen in the model learned at 90% label coverage. This model was unable to restore question marks because its training set, whose labels were formed from the first labels sorted by the frequency of occurrence, did not include any class containing a question mark. The lack of decrease in performance on this task shows that the current deep models are capacious enough that increasing the range of labels (and thus both the resolution and the range of predicted punctuation marks) does not carry a cost in terms of a decrease in a model quality on the prediction of the more salient marks.

6 Results for Polish

The results for the Polish language are presented in Table 7. In general, Polish turned out to be

Training labels	Text punctuated by model
$\text{Base}_{\text{Alam}}$ (4 classes)	as a result of many inspections, payments of own resources and interest were demanded for agriculture as a whole 49 8 billion in 2006, the court found a marked reduction in the estimated overall level of error.
90^{th} percentile (7 classes)	as a result of many inspections, payments of own resources and interest were demanded for agriculture as a whole 49 8 billion in 2006, the court found a marked reduction in the estimated overall level of error.
95^{th} percentile (13 classes)	as a result of many inspections, payments of own resources and interest were demanded for agriculture as a whole (49 8 billion in 2006, the court found a marked reduction in the estimated overall level of error.
99^{th} percentile (24 classes)	as a result of many inspections, payments of own resources and interest were demanded for agriculture as a whole (49.8 billion in 2006), the court found a marked reduction in the estimated overall level of error.
GOLD	as a result of many inspections, payments of own resources and interest were demanded. for agriculture as a whole - €49.8 billion in 2006 - the court found a marked reduction in the estimated overall level of error.

Table 4: Example predictions (on an excerpt from test set) from the best model for English (RoBERTa) trained on a different number of training labels.

Model	Label Set	P	R	f1	mRS
BiGRU Tilk and Alumäe (2016)	$\text{Base}_{\text{Tilk}}$	79.65	60.06	68.48	60.30
	Reduced	76.17	56.28	64.73	57.23
	90^{th}	78.68	67.72	72.79	68.02
	95^{th}	77.51	70.07	73.60	70.37
	99^{th}	77.97	72.94	75.37	73.27
Bert-base + LSTM + Aug Alam et al. (2020)	$\text{Base}_{\text{Alam}}$	84.58	68.28	75.57	68.50
	Reduced	84.61	68.26	75.56	68.48
	90^{th}	85.35	77.45	81.21	77.70
	95^{th}	84.78	80.30	82.48	80.59
	99^{th}	85.52	82.18	83.82	82.53
RoBERTa-base + LSTM + Aug Alam et al. (2020)	$\text{Base}_{\text{Alam}}$	85.57	68.63	76.16	68.85
	Reduced	84.15	68.41	75.47	69.49
	90^{th}	86.70	77.48	81.83	77.74
	95^{th}	86.49	79.58	82.89	79.85
	99^{th}	85.87	82.77	84.29	83.14

Table 5: Performance of the models on comprehensive punctuation restoration task for the English language. Each model was trained under multiple subsets of labels. The base label set corresponds to the label subset used in the original model's implementation. The reduced label set corresponds to models trained on the reduced dataset.

		COMMA			PERIOD			QUESTION			TOTAL		
Model	L	P	R	f1	P	R	f1	P	R	f1	P	R	f1
BiGRU	$\text{Base}_{\text{Tilk}}$	76.3	62.5	68.7	83.9	78.0	80.9	76.8	63.3	69.4	83.6	75.6	79.3
	Reduced	73.9	65.3	69.4	84.7	79.1	81.8	77.8	54.3	63.9	83.5	74.4	78.3
	90^{th}	73.8	67.8	70.7	82.5	80.4	81.4	0*	0*	0*	63.5	61.7	62.6
	95^{th}	71.2	70.8	71.0	83.6	78.6	81.0	75.0	63.5	68.8	81.9	77.8	79.7
	99^{th}	72.0	70.5	71.2	82.9	84.4	83.7	75.8	61.9	68.1	82.3	78.8	80.3
BERT-base + LSTM + Aug	$\text{Base}_{\text{Alam}}$	79.9	80.1	80.0	92.2	89.3	90.8	88.1	79.5	83.5	89.7	86.9	88.2
	Reduced	80.0	80.0	80.0	92.2	89.3	90.8	87.9	79.7	83.6	89.7	87.0	88.3
	90^{th}	79.8	80.3	80.0	93.0	88.3	90.6	0*	0*	0*	67.8	66.8	67.3
	95^{th}	78.9	81.8	80.3	92.8	88.8	90.8	88.7	79.4	83.8	89.8	87.2	88.4
	99^{th}	80.5	79.7	80.1	92.3	93.6	93.0	87.3	80.0	83.5	89.7	88.0	88.8
RoBERTa -base + LSTM + Aug	$\text{Base}_{\text{Alam}}$	81.0	80.3	80.7	93.0	90.0	91.5	90.8	78.3	84.1	90.9	86.9	88.8
	Reduced	81.3	80.2	80.8	93.7	94.2	94.0	91.8	80.1	85.6	91.4	88.4	89.8
	90^{th}	82.2	79.0	80.6	93.2	90.2	91.7	0*	0*	0*	68.4	67.0	67.7
	95^{th}	81.8	79.2	80.5	93.4	89.8	91.6	89.4	80.7	84.8	90.8	87.2	88.9
	99^{th}	80.8	81.0	80.9	93.9	93.4	93.8	91.4	78.8	84.6	91.2	88.0	89.4

Table 6: Comparison of models under task with a reduced set of labels for the English dataset. We can see that adding more labels did not negatively impact the model's performance in the base formulation of the task. (*) The zero values are caused by the fact that a question mark was not included in the 90^{th} percentile of all punctuations.

a slightly easier punctuation restoration task as a whole. The best f1 score obtained for Polish was 85.93 as compared to 84.29 obtained for English. In the case of Polish, the effect of adding subsequent classes on increasing recall was smaller (although still relatively large). In the BERT model, adding more classes strictly decreased the average prediction precision, but in the recursive model (BiGRU), no clear trend was observed. This is somewhat opposite to the results obtained on the English set. Similarly to English, we also tested whether models trained on the larger label set would decrease in performance on the baseline task of 4 classes. The results of the models on this task are presented in Table 8. There was no noticeable effect of increasing the number of labels on the quality of the model in predicting basic labels. For Polish, we found that the task of restoring commas was easier, while that of restoring question marks was much more difficult. We suspect that this might be rooted in the structure of language because, in the Polish language, one can often come across question structures that differ from the indicative sentence only by the question mark at the end - e.g., it's common to use structures like "jesteś szczęśliwy?" ("you are happy?") rather than "czy jesteś szczęśliwy?" (that would resemble "are you happy?"). However, to make a definite statement, it would be necessary to conduct further research in this area, especially since the basis of BERT's methods is a language model, which for obvious reasons was pre-trained on different sets for each of the two languages.

7 Conclusion and Future Work

In our work, we have shown that token classifier models are able to restore a much larger range of punctuation than it is done in most other reported researches. Our experiments show that such an increase in coverage can be achieved without a drop in quality for key punctuation marks. We have also shown that this effect is not limited to English, and we have obtained very similar results in Polish — a language with much more complex morphology. Additionally, the advantage of the approach with automatic generation of a set of labels from the data is that we are also able to predict the composition of punctuation marks. In further work, it would be of great benefit to investigate what effect reproducing a wide range of punctuation would have on text readability for people compared to reproducing only the basic characters. It would also be interesting to perform a comparative study of how token classifier models perform in the task of reproducing broad punctuation compared to sequence-to-sequence models, such as Bart (Lewis et al., 2020) or T5 (Raffel et al., 2020) for which such behavior would be natural. We also plan to take part in the

Model	Training Task	P	R	f1	mRS
BiGRU Tilk and Alumäe (2016)	$\text{Base}_{\text{Tilk}}$	84.52	69.73	76.42	70.09
	Reduced	82.44	67.61	74.29	68.59
	90^{th}	84.00	68.43	75.42	68.78
	95^{th}	83.89	70.97	76.89	71.37
	99^{th}	84.11	73.98	78.72	74.57
Bert-base + LSTM + Aug Alam et al. (2020) Kłeczek (2020)	$\text{Base}_{\text{Alam}}$	89.64	75.32	81.86	75.60
	Reduced	87.62	75.55	81.15	76.67
	90^{th}	88.49	78.83	83.38	79.18
	95^{th}	87.98	80.78	84.22	81.17
	99^{th}	87.71	84.22	85.93	84.82

Table 7: Performance of the models on comprehensive punctuation restoration task for the Polish language. Each model was trained under multiple subsets of labels. The base label set corresponds to the label subset used in the original model's implementation. The reduced label set corresponds to models trained on the reduced dataset.

		COMMA			PERIOD			QUESTION			TOTAL		
Model	L	P	R	f1	P	R	f1	P	R	f1	P	R	f1
BiGRU	$\text{Base}_{\text{Tilk}}$	87.1	77.0	81.7	82.7	79.5	81.1	66.3	47.3	55.2	83.4	75.6	79.0
	Reduced	86.2	78.8	82.4	82.8	82.8	82.8	68.5	47.1	55.8	83.8	76.8	79.8
	90^{th}	86.6	76.7	81.3	81.8	77.2	79.5	0*	0*	0*	66.4	63.2	64.7
	95^{th}	86.6	77.8	82.0	81.8	80.2	81.0	66.2	47.0	54.9	83.0	75.9	79.0
	99^{th}	87.1	77.2	81.8	81.8	83.1	82.4	61.7	49.4	54.8	82.1	77.1	79.3
BERT-base + LSTM + Aug	$\text{Base}_{\text{Alam}}$	89.2	86.1	87.6	92.1	89.8	91.0	83.5	77.8	80.5	90.8	88.2	89.5
	Reduced	89.2	86.1	87.6	92.1	89.8	91.0	83.5	77.8	80.5	90.8	88.2	89.5
	90^{th}	88.0	87.4	87.7	92.0	89.3	90.7	0*	0*	0*	69.6	68.9	69.2
	95^{th}	87.9	87.6	87.8	91.4	90.3	90.9	80.5	79.1	79.8	89.6	89.0	89.3
	99^{th}	87.0	88.5	87.7	92.2	92.1	92.2	85.9	74.5	79.8	91.0	88.4	89.6

Table 8: Comparison of models under original task with a reduced set of labels for Polish dataset. (*) The zero values, similarly to results for English, are caused by the fact that a question mark was not existing in the 90^{th} percentile of all punctuations.

PolEval 2021[5] shared task, concerning punctuation restoration from read text in Polish. In contrast to the problem analyzed here, the data sets will contain acoustic information, e.g. one that could allow determining the duration of gaps between words.

Acknowledgments

Financed by the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN - Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19.

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.

Doug Beeferman, A. Berger, and J. Lafferty. 1998. Cyberpunc: a lightweight punctuation annotation system for speech. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 2:689–692 vol.2.

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.

Michel Marie Deza and Elena Deza. 2009. *Encyclo-*

[5] http://poleval.pl/tasks/

pedia of Distances. Springer Berlin Heidelberg, Berlin, Heidelberg.

Daniel Hládek, Ján Staš, and Stanislav Ondáš. 2019. Comparison of recurrent neural networks for slovak punctuation restoration. In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 95–100.

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.

Dariusz Kłeczek. 2020. Polbert: Attacking polish nlp tasks with transformers. In *Proceedings of the PolEval 2020 Workshop*, pages 79–88. Institute of Computer Science, Polish Academy of Sciences.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xinxing Li and Edward Lin. 2020. A 43 language multilingual punctuation prediction neural network model. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1067–1071. ISCA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 3047–3051. ISCA.

Máté Ákos Tündik, György Szaszák, Gábor Gosztolya, and András Beke. 2018. User-centric evaluation of automatic punctuation in ASR closed captioning. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 2628–2632. ISCA.

Vincent Vandeghinste, Lyan Verwimp, Joris Pelemans, and P. Wambacq. 2018. A comparison of different punctuation prediction approaches in a translation context. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 269–278.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30 of *NIPS'17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

Shudong Yang, Xueying Yu, and Y. Zhou. 2020. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, pages 98–101.

Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Cunhang Fan. 2020. Adversarial transfer learning for punctuation restoration. *CoRR*, abs/2004.00248.

Marta Łockiewicz and Martyna Jaskulska. 2017. Polish as l1, english as l2: the linguistic transfer impact on second language acquisition stemming from the interlingual differences: implications for young learners education. *Problemy Wczesnej Edukacji*, 37(2):68–76.

Piotr Żelasko, Piotr Szymański, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. 2018. Punctuation prediction model for conversational speech. pages 2633–2637.

Appendix A: List of generated labels for English dataset

	Label	Occurrences (%)	Percentile
1	,[S]	44.97	44.97
2	.[S]	32.42	77.39
3	-	4.46	81.85
4	’	2.86	84.71
5	[S]-[S]	2.37	87.08
6	)[S]	2.26	89.34
7	:[S]	1.28	90.61
8	(	1.22	91.84
9	’[S]	0.93	92.76
10	?[S]	0.90	93.66
11	[S](	0.79	94.46
12	;[S]	0.79	95.25
13	[S]’	0.76	96.01
14	.[S]-[S](	0.55	96.56
15	%[S]	0.54	97.10
16	.[S]-[S]	0.49	97.59
17	/	0.43	98.02
18	.	0.30	98.32
19	’.[S]	0.23	98.55
20	’,[S]	0.14	98.69
21	),[S]	0.13	98.82
22	).[S]	0.12	98.94
23	![S]	0.11	99.05

Table 9: List of labels that together cover at least 99% of punctuation cases for English version of the dataset. Spaces were replaced with [S] for readability.

Appendix B: List of generated labels for Polish dataset

	Label	Occurrences (%)	Percentile
1	,[S]	51.21	51.21
2	.[S]	31.42	82.62
3	[S]-[S]	3.00	85.63
4	![S]	2.13	87.76
5	)[S]	1.94	89.70
6	:[S]	1.15	90.85
7	(	0.96	91.80
8	-	0.83	92.64
9	?[S]	0.82	93.46
10	[S](	0.79	94.25
11	[S]"	0.76	95.01
12	;[S]	0.69	95.69
13	.[S]-[S](	0.42	96.12
14	/	0.40	96.51
15	.[S]-[S]	0.36	96.87
16	”[S]	0.35	97.22
17	”.[S]	0.29	97.51
18	%[S]	0.24	97.75
19	[S]%[S]	0.20	97.95
20	”,[S]	0.18	98.14
21	,	0.17	98.30
22	),[S]	0.13	98.44
23	).[S]	0.13	98.56
24	.	0.10	98.67
25	’	0.10	98.77
26	:[S]"	0.10	98.87
27	.,[S]	0.09	98.96
28	[S]([S]	0.07	99.03

Table 10: List of labels that together cover at least 99% of punctuation cases for Polish version of the dataset. Spaces were replaced with [S] for readability.