NATURAL LANGUAGE PROCESSING 2022

PUNCTUATION RESTORATION MILESTONE 1.

May 13, 2022

Elżbieta Jowik (298821), Agata Kaczmarek (305751), Agata Makarewicz (298827)

1 Introduction / Project description

The purpose of this project is to restore punctuation in the Automatic Speech Recognition (abbr. ASR) of texts read out loud. It is a task from PolEval 2021 competition, which aims to create and compare natural language processing tools for the Polish language. Speech transcripts generated by ASR systems typically do not contain any punctuation or capitalization, which typically affects the general clarity of the output text. The primary purpose of punctuation restoration (abbr. PR) is to improve the legibility of such text. The task itself is hard mainly because punctuation rules can be ambiguous even for originally written texts, and the very nature of naturally-occurring spoken language makes it difficult to identify clear phrase and sentence boundaries.

The data source provided for the task is WikiPunct - a crowdsourced text and audio data set of Polish Wikipedia pages read out loud by Polish lectors. The data set consists of two parts: conversational (WikiTalks) and informational (WikiNews). Texts are original and may contain some user-induced errors and bias, therefore, punctuation in them should be approached with some reservation. The texts were read out by over a hundred different speakers. Therefore, after proper preprocessing, we are able to use both text-based and speech-derived features to identify punctuation marks. Original texts with punctuation were forced-aligned with recordings and used as the ideal ASR output.

According to the organizers of the competition, we consider 8 types of punctuation marks:

- fullstop (`.`)
- comma (`,`)
- question mark (`?`)
- exclamation mark (`!`)
- hyphen (`-`)
- colon (`:`)
- ellipsis (`...`)
- semicolon (`;`)

Additionally, quotes (`"`) and brackets (`()`) are present in the texts, however they are not considered as punctuation marks in this task.

2 Literature overview

The subject of automatic punctuation restoration was raised by Nagy et al. [4]. The research was conducted for both English and Hungarian languages, and the best versions of the models achieved F_1 score of about 80. Researchers found out, that the most efficient systems used for restoring punctuation, were using both prosodic and lexical features of the data. They used a pre-trained BERT model [1], which helped them to raise the accuracy of the model. The authors decided to create only four classes for their punctuation problem: empty (no punctuation after the word), comma, period and question. Other punctuation marks were too rare, so they changed (!) for (.) then (;) for (,) and (?) for (.), to still be able to restore at least some similar mark. The authors noted the fact, that the data set was unbalanced - class empty had more than 85% of all observations, so the F_1 score was calculated without this class. During the research, other versions of BERT algorithms were used - supporting more languages.

Introduction to the BERT model was done by Devlin et at. [1], defining it as a language representation model named Bidirectional Encoder Representations from Transformers. A pre-trained model is created using unlabeled data for various tasks, whereas for finetuning, the model should be initialized with pre-trained parameters. Then using labelled data, it should be finetuned, and used for solving problems. BERT achieved one of the best results at the time of creation. For the Polish language, multilingual BERT is used.

Woliński [7] introduced a Practical Tool for the Morphological Analysis of Polish - Morfeusz. It was created using a book about inflexion in Polish, which includes information about all possible endings of words in Polish ("Schematyczny indeks a tergo polskich form wyrazowych", Tokarski Jan). In Morfeusz, instead of including all possible words in the dictionary, creators decided to split some words into parts (eg. they split "szczyt polsko-czesko-węgierski" into parts). Thanks to this tool, users can get the result of the morphological analysis, lemma (set of forms, that describes the same object) and tag (values of grammatical categories, that specify the form).

For this project with Punctuation Restoration, we are using both textual and audio data. Using openSMILE tool, introduced by Eyben et al. [2], speech can be processed. The audio analysis is important for Automatic Speech Recognition (ASR) tasks. With help of this toolkit, low-level descriptions of audio features can be found, which can be used not only for speech recognition but also for analyzing heart rate and EEG. The detailed instructions, on how to understand the most important (for our research) of those features, [5] and [6] can be mentioned. In the first one, Sandhya et al. explain and use properties of Spectral Features for Speaker Recognition with emotional bias. The authors explain, that eg. the MFCC feature varies, when the speaker is under influence of various emotions. Additionally, researchers can get relevant information from speech signals, including language, emotion, gender or age. In the second one, Teixeira et al. [6] explain how they used fundamental frequency, jitter, shimmer and HNR parameters for their research on diagnosing pathologies of the larynx.

The Punctuation Restoration task was introduced as the PolEval2021 Task. There were various contributions made and some of them ended as short articles describing different solutions to this problem. Wróbel and Zhylko [8] created a solution, which was scored first. Their solution includes using small preprocessing and using a pre-trained HerBERT model with added one new layer. Both Marcińczuk [3] and Ziętkiewicz [9] created alternative solutions for this problem, also mentioned in the group of the best ones.

3 Data sets overview

There are two types of data provided: text and audio. Text data is provided in `.tsv` format (i.e. Tab Separated Values), whereas audio data is provided in `.wav` files and additionally, in the form of force-aligned transcripts with timestamps. The explanatory data analysis we conducted was oriented to the training data set.

3.1 Text data

The training data set consists of 800 records, each one corresponding to one entry from WikiNews or WikiTalks services. The analysis was performed on the expected output of our model in order to understand the dependencies determining the placement of punctuation marks we are to restore.

Firstly, we tokenized provided data into sentences. On average, one-sentence consists of 14 words, however, there are a few obvious outliers, for instance, a sentence containing 272 words. We also analyzed the frequency of occurrences of the considered punctuation marks. It turned out that the classes (types of punctuation marks) are highly imbalanced. There is a significantly bigger number of occurrences of commas and fullstops than the rest of the considered marks. It is explainable, given that each sentence is usually closed with a fullstop, and a comma often appears next to particles or linking words.

Table 1: Several occurrences of each of the considered punctuation marks in the training data set.

Fullstop	Comma	Question mark	Exclamation mark	Hyphen	Colon	Ellipsis	Semicolon
10478	10132	797	118	2448	913	3	90

Moreover, we analyzed which words appear most often after the given punctuation mark. We visualized the results using word clouds, some of which are presented in Figure 1. In the case of commas, we can observe that the most frequent words are particles (*że, czy*), linking words (*ale, bo*) and pronouns (*który, która*). These are the words before which in the Polish language we always place a comma. Therefore, there are some ground rules which will be useful in our task. On the other hand, in the case of hyphens, there are no words that would dominate in terms of frequency. However, a large majority of all the words appearing after this punctuation mark are verbs in past tense and in the third person (for instance *powiedział*), which would indicate that the hyphen is mostly used for mentioning someone's statement. Such dependency will also prove useful, because there is a tool called *Morfeusz* (Section 4) which can recognize the characteristics of the word in a sentence, including the conjugation. Such information will let us construct rules based on which we will build our baseline model (Section 4).

Figure 1: Words occurring after comma (on the left) and hyphen (on the right). The size of the word corresponds to the number of its occurrences.

3.2 Audio data

3.2.1 Forced-aligned transcriptions

Force-aligned transcriptions of the original texts are used to approximate Automatic Speech Recognition (abbr. ASR) output. Files are provided in the `.clntmstmp` format and correspond to the `.wav` audio files, which contain the recordings of volunteers reading the texts out loud. Exploratory Data Analysis proved that there are some recordings, for which the transcription with timestamps is not provided. We have at our disposal 793 training transcription files, while there are 800 audio files available (as well as 800 records in the corresponding `.tsv` file). Moreover, in the missing 7 files, there are all 3 observations containing ellipsis and as a result, in the considered training set there is no observation corresponding to one of the target classes. Data is given in the following format:

`(timestamp_start,timestamp_end) word`

Timestamps are provided in milliseconds and they let us analyse the pauses between the words that the reader introduced while reading the texts. Typically bigger pauses correspond to the presence of punctuation marks - they occur between the sentences, after the comma or before the hyphen. Therefore, such information can be really helpful in the punctuation restoration task. We analyzed the distribution of the length of the pauses appearing before and after the considered punctuation marks. Two marks were omitted in the analysis: ellipsis because all of its occurrences in the training data set are present in the records of which we do not have a transcript provided; and semicolon, because according to the organizers of the competition, it was not included in the final evaluation. Additionally, before the mentioned analysis we dropped outliers from the training set, i.e words before or after which occurred a pause longer than 4.5 seconds. The threshold was adjusted empirically, by analysing the pauses length distribution plots. The distributions of the length of the pauses appearing after the punctuation marks while reading the texts are far more varied than in the case of the pauses appearing before them. This implies that the feature describing the length of the pause before the mark may be a valuable attribute in our task.

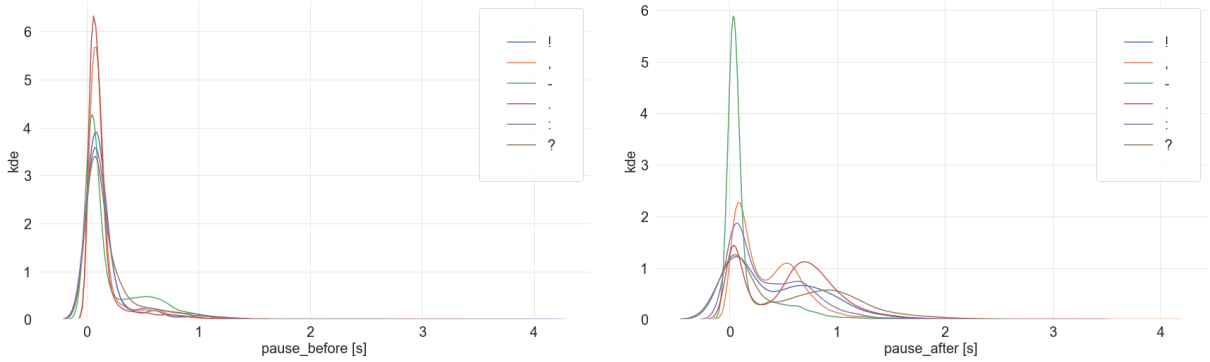

Figure 2: The distributions of the length of the pauses in the recordings (transcripts) appear before the punctuation marks (on the left) and after them (on the right). The length of the pauses is given in seconds.

3.2.2 Audio files (`.wav`)

According to the PolEval 2021 task description in the audio dataset, 2 subtypes of recording can be distinguished:

- randomly selected WikiNews (80% of the training set) with the word count above 150 words and smaller than 300 words;
- randomly selected WikiTalks (20%) with words the count above 150 words but smaller than 300 words and at least one question mark

The core objective of audio data exploratory analysis were features extraction, evaluation of their informativeness from the task under consideration point of view and assessment of whether the available hardware resources will allow them to be processed.
At the beginning of work with audio files *.wav, we decided to do an Explanatory Data Analysis, to get to know the given data. During this work, we were working on 800 recordings, which were transformed into a frame, having as a column various audio features, which can be taken from the above-mentioned recordings.

In the beginning, we decided to look at audio volume. We decided to check how big are our data, to consider if we should abandon some of them already at the beginning. Overall, most

of the files have sizes between 2.5-4 MB, distribution seems to be unimodal and right-skewed. No unambiguous signs that we should reject some of them. All 800 files have a volume of 2.39GB. Afterwards, the extraction of features has been done. It was done only once, as it was time-consuming (took over 4 hours) and we decided to save the results of an extraction function $smile.process\_file$ to *.csv files. The loading times of separate files can be seen below and they are connected to the volume of data, which are loaded.

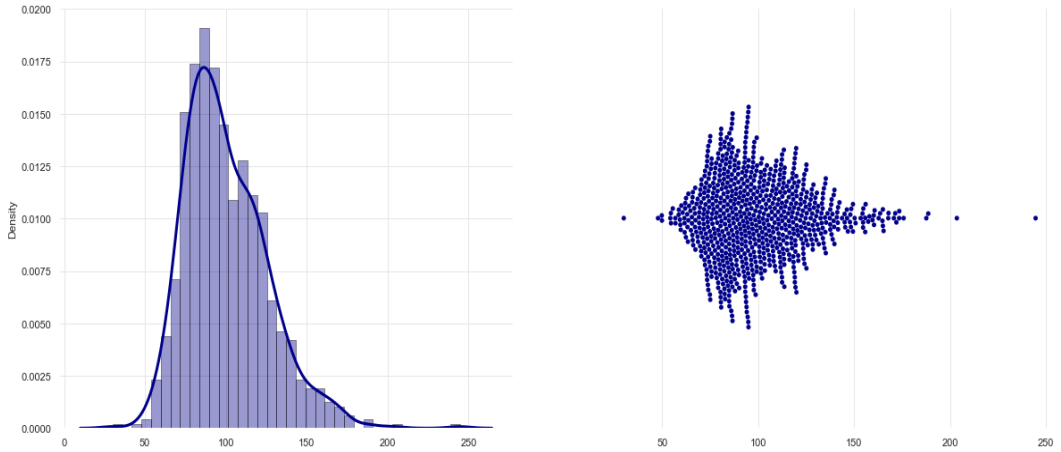

Figure 3: Plot showing the distribution of files lengths on the left and the swarm plot of it on the right

Additionally, it can be seen, that in the features set (all 800 recordings) there are 0 columns, that are zeroed in all of the recordings. There are 33 columns (features), that do not have any zero values in any of the recordings. In features describing jitter and shimmer (voice quality features, describing amplitude/ variation between pitches) more zeroes were occurring, which seems to be good, taking into consideration the definition of those features. For features describing Spectral Features like centroid or energies, or MFCC describing characteristics, the number of zeros was smaller, which also seems logically understandable.

For further work, we chose only one representative recording and we performed all work on it. The recording was chosen randomly from recordings having the most frequent length. Firstly, to see how does the data look like, we plotted the waveform of signals and their autocorrelation. Looking only at one of the recordings it could not be seen, but as we, during the work, plotted more of those, some of the patterns could be seen. We think, that these patterns are worth investigating in feature engineering, to see the predictive power of the course of autocorrelation.

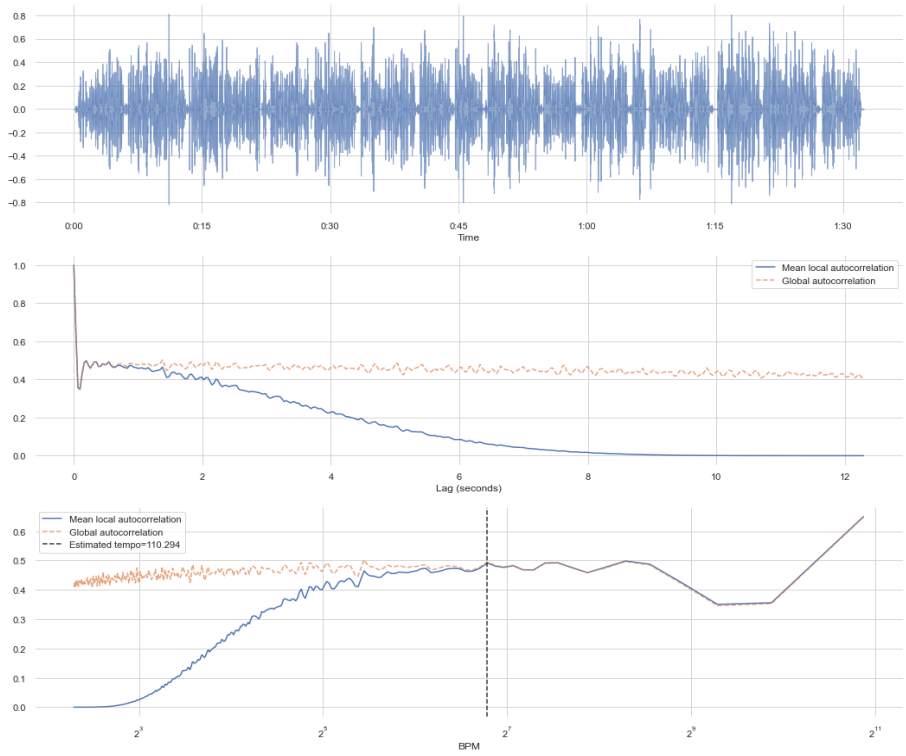

Figure 4: Plots showing the waveform of the signals and their mean and global autocorrelation

Afterwards, few chromatograms were done, as they are the measure of sound intensity. Values of those are in various moments significantly different, and the knowledge about it could be used during the modelling phase. While comparing energy and spectrograms, strong relationships can be seen between them. The energy seems to be a good reflection of this course.

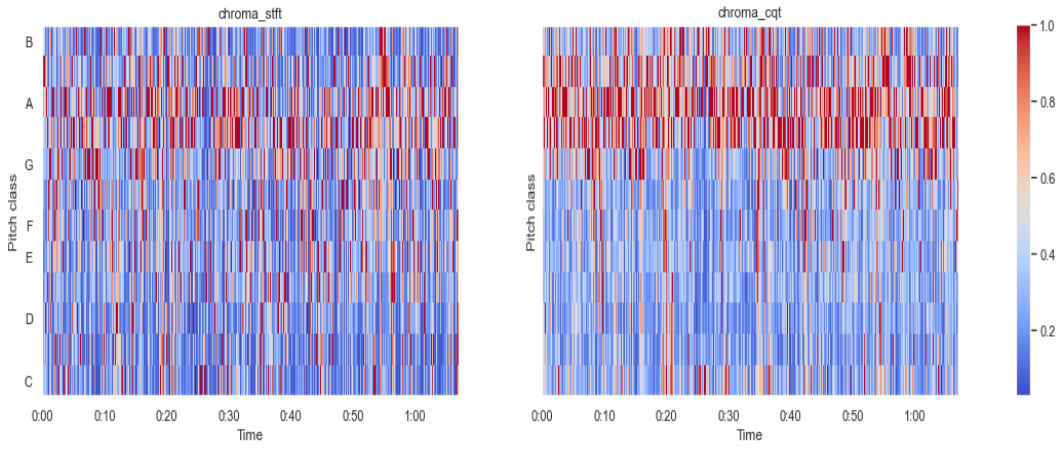

Figure 5: Plots showing the chromatograms for the data

Last but not least, we decided to concentrate on the MFCC feature, which captures characteristics, that gives good performance for, for instance, speech recognition. There is no single definition of the mel scale, so we used librosa library, which has a few implementations of this feature and tried to compare them on our plots.

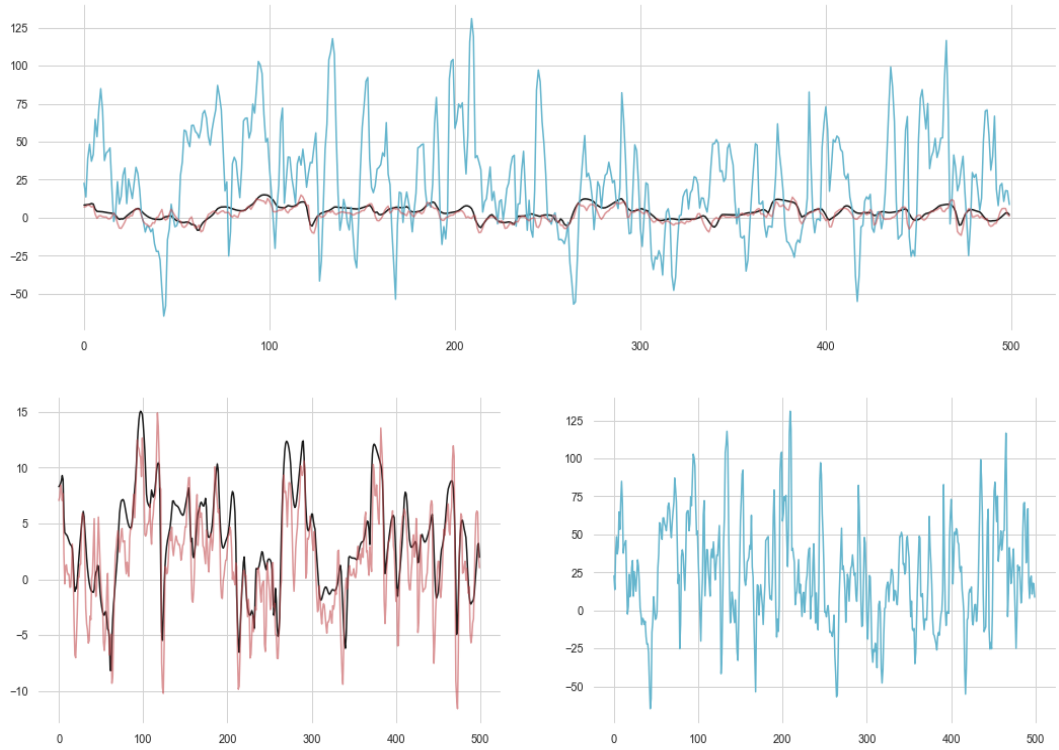

Figure 6: Plots showing the comparison between various implementations of MFCC feature in librosa library

4 Solution concept and future work

After gaining insight into the problem from the literature and conducting the explanatory data analysis on provided data sets, we plan to perform the following steps in the next phases of the project.

1. **Cleansing the data.** Includes filtering out the anomalies and outliers identified statistically as well as identifying the records for which we do not have the transcripts provided and probably dropping them out of our analysis.

2. **Feature engineering.** Includes morphological analysis conducted using the *Morfeusz* tool. It identifies the lemma, part of speech, values of the number, case and gender (if applicable) and assigns each word an appropriate tag according to the identified values. This tool will enable us to verify whether the placement of the considered punctuation marks is determined by the part of speech or form of the neighbouring word, which would be helpful in building the baseline model.

3. **Baseline model.** Includes creating a baseline model based on the predefined rules identified during explanatory data analysis and feature engineering. For that purpose, for instance, a dictionary of words before which we always insert a comma will be created. Rules will be constructed in the following schema (pseudocode):

```
if word.partofspeech == 'verb' and word.number == 3 then insert "-"
                if word in comma_dict then insert ","
```

4. **Final model.** Includes creating a pre-trained HerBERT model, which will be created based on the transformers package (tokenizer and model). The idea of the solution is similar to the one Wróbel et al. [8] did for their solution for the Punctuation Restoration task during

PolEval2021. However, at least in the article, there was no information about using audio data. We plan, that this data will be part of our solution. However, the future challenge in this solution will be using the sliding window for a tokenized sequence of data, so that data will not get lost.

5. **Evaluation of the solution.** Final results will be evaluated in terms of *precision*, *recall*, and *F1* scores for predicting each punctuation mark separately. The global score per punctuation sign p will be calculated using a micro-averaged version of mentioned metrics and the final scoring metric will be a weighted average of global scores:

$$\frac{1}{N} \sum_{p \in Punctuation} support(p) * avg_{micro} F_1(p) \tag{1}$$

References

[1] Devlin, J., Chang, M.W., Lee, K and Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018

[2] Eyben, F., Wollmer, M. and Schuller, B. "openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor", 2010

[3] Marcińczyk, M. "Punctuation Restoration with Ensemble of Neural Network Classifier and Pre-trained Transformers", 2021

[4] Nagy, A., Bial, B. and Acs, J. "Automatic punctuation restoration with BERT models", 2021

[5] Sandhya, P., Spoorthy, V., Koolagudi, S.G. and Sobhana, N.V. "Spectral Features for Emotional Speaker Recognition", 2020

[6] Teixeira, J. P., Oliveira, C. and Lopes, C. "Vocal Acoustic Analysis - Jitter, Shimmer and HNR Parameters", 2013

[7] Woliński, M "Morfeusz - a Practical Tool for the Morphological Analysis of Polish", 2006

[8] Wróbel, K. and Zhylko, D. "Punctuation Restoration with Transformers", 2021

[9] Ziętkiewicz, T. "Punctuation Restoration from Read Text with Transformer-based Tagger, 2021