Automatic Grammar Correction of Commas in Czech Written Texts: Comparative Study*

Jakub Machura[1][0000−0002−6623−3064], Adam Frémund[2][0000−0001−8780−6629], and Jan Švec[2][0000−0001−8362−5927]

[1] Department of Czech Language, Masaryk University, Brno, Czech Republic
machura@phil.muni.cz
[2] Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic
{afremund,honzas}@kky.zcu.cz

Abstract. The task of grammatical error correction is a widely studied field of natural language processing where the traditional rule-based approaches compete with the machine learning methods. The rule-based approach benefits mainly from a wide knowledge base available for a given language. On the contrary, the transfer learning methods and especially the use of pre-trained Transformers have the ability to be trained from a huge number of texts in a given language. In this paper, we focus on the task of automatic correction of missing commas in Czech written texts and we compare the rule-based approach with the Transformer-based model trained for this task.

Keywords: Grammatical error correction · Linguistic rules · Transfer learning.

1 Introduction

Sentence punctuation is a very important linguistic feature that helps the reader better understand the complex text flow. While the sentence separating punctuation (such as full stops and question marks) is crucial for marking the basic sentence units in the text, the intra-sentence punctuation (mostly commas) helps to structure the sentence on syntactic and semantic levels [18,2].

If we focus on tasks where the missing or wrongly-inserted punctuation is restored, two main tasks arise (1) punctuation restoration in speech transcripts from automatic speech recognition and (2) grammatical error correction in written texts. The first task is widely studied because the restored punctuation dramatically improves the readability of the recognized transcript. The current methods use sequence-to-sequence mapping or token classification. The neural networks with recurrent units (such as LSTM or GRU) or self-attention mechanism (mostly Transformer) are used. The models can use only the lexical information [4,23] or also the acoustic features extracted from speech [12,9].

In his paper, we focus on the grammatical error correction in written text. Also, this task is widely studied in many languages (for example [3,7]). We present a comparison of

* This work was supported by the project of specific research *Lexikon a gramatika češtiny II - 2022* (Lexicon and Grammar of Czech II - 2022; project No. MUNI/A/1137/2021) and by the Czech Science Foundation (GA CR), project No. GA22-27800S.

Petr Sojka, Aleš Horák, Ivan Kopeček and Karel Pala (Eds.): TSD 2022, LNAI 13502, pp. 107–118, 2022.
This is preprint prepared by Proceedings editor for Springer International Publishing Switzerland.

two approaches for a specific task of automatic correction of commas in Czech written texts. We first introduce the typology of rules for writing commas in Czech language (Sec. 2). Then we describe the rule-based (Sec. 3.1) and the Transformer-based (Sec. 3.2) approaches which are evaluated on the same datasets described in Sec. 4. Sec. 5 presents the results of experimental evaluation and Sec. 6 concludes the paper.

2 Writing Commas in Czech language

Regarding punctuation, the main attention is given to writing commas. The comma is the most frequent punctuation mark not only for Czech but also for Slovak, English or German (see [23], [5]). Traditionally, writing commas in Czech is part of the orthography. However, rather than orthographic rules at the level of words, these are rules of higher syntactic units. The rules for writing commas are codified in *Pravidla českého pravopisu* (Rules of Czech Orthography) [1] and in *Akademická příručka českého jazyka* (Academic Handbook of the Czech Language) [19]. Thus, we would say that knowing how to use commas is a part of "knowing how to write".

Nunberg [17] recognizes two main classes of commas. He sees a difference between a comma which separates structures at the same level (the *separator comma*), and a comma which delimits the boundaries between syntactic structures at different levels (the *delimiter comma*). The *separator comma* is inserted between members of coordinated sentence elements of the same type (e.g., multiple subject, object, predicate etc.) or of coordinated clauses within the sentence (asyndeton, or parataxis). The *delimiter comma* marks the boundaries between the main clause and the subordinate clause (hypotaxis), vocatives, or parenthetical expressions. In view of this classification, we can probably think of the existence of a third type of comma which cannot be obviously assimilated to either of these others. Its presence affects the meaning of the utterance (see Sec. 2, D.ii).

2.1 Typology of the comma insertion place

To formalize the linguistic rules, it was necessary not only to consider the classification of commas according to which textual categories they separate but also to specify the place (boundary) in the sentence structure where the comma is inserted. Such a typology was first outlined in [7] and it is extended and described in detail below. In addition, we created a small random sample of 183 sentences and classified the commas in the text according to the established typology (see Tab. 1).

A. The comma precedes a connective
A connective (conjunction, relative pronoun, or relative adverb) or group of connectives indicates:

(i) the boundary between main clauses or multiple elements of a sentence which are not in simple coordination relation (see "Koordinace" in [10]):
Pozvali jsme Karla, ***ale*** *přišel Petr. (We invited Charles,* ***but*** *Peter came.)*
Buď *přijedou dnes večer,* ***nebo*** *zítra ráno. (They will come* ***either*** *tonight* ***or*** *tomorrow morning.)*

(ii) the boundary between the main clause and subordinate clause (see "Souvětí" in [10]):
Otec neví, ***na jaký*** *úřad má jít. (Father does not know* ***what*** *bureau to go in.)*
(iii) apposition with an additional modification:
Společnost dosáhla nového vrcholu v zisku, ***a to*** *220 milionů korun. (The company has reached a new peak in profit,* ***namely*** *CZK 220 million)*

B. The comma is located between two clauses without the (close) presence of a connective

(i) Connections of clauses in asyndetic structures (see "Asyndeton" in [10]):
Petr má rád červené víno, jeho žena miluje bílé. (Peter likes the red wine, his wife loves the white one.)
(ii) A connective usually stands on the left side of the subordinate clause, and the subordinate clause is separated asyndetically from the right side:
Auto, které stálo celou noc před dome, se rozjelo. (A car, which was standing in front of the house the whole night, moved off.)
(iii) And finally, sentences containing direct speech or quotation:
„Musíte jít na operaci," řekl lékař. ("You must have surgery!" said the doctor.)

C. The comma separates individual components of multiplied syntactic structure

(i) Multiple sentence elements or enumeration. We assume that multiple sentence elements group together words that agree in some grammatical category – e. g. part of speech or case. If it has more than two members, most often only the last two members are separated by a conjunction. Other members are separated asyndetically, only by the comma (see "Koordinace" in [10]):
Mezi oblíbené turistické destinace letos patří Španělsko, Francie, Itálie a Chorvatsko. (This year Spain, France, Italy, and Croatia belong to popular tourist destinations.)
(ii) The apposition – construction of two elements, usually noun phrases, which are placed side by side and the second element somehow describes the first element – is another type of structure where is obligatory to put the comma:
Sněžka, nejvyšší hora České republiky (Sněžka, the highest mountain of the Czech Republic)

The main difference between (i) and (ii) is characterized by the fact that components of (i) refer to different entities. However, components of (ii) relate fully or partially to a single entity (see "Apozice" in [10]).

D. The comma might but might not be inserted or affects the meaning of the utterance

(i) The writer sometimes has the option of whether or not to write a comma. These include parentheses or phrasal idioms:
Zítra(,) bohužel(,) přijít nemohu. (Unfortunately, I can't come tomorrow.)

(ii) In some cases, the insertion of the comma changes the meaning of the utterance. The typical examples in Czech are a restrictive attribute and nonrestrictive attribute (see "Přívlastek" in [10]):
Jídlo, koupené v obchodních domě, jsem uložil do ledničky. (I put the food, which was bought in the mall, into the fridge – all food was bought in the mall and I put all this food in the fridge.*)*
Jídlo koupené v obchodním domě jsem uložil do ledničky. (I put food bought in the mall into the fridge – only food which was bought in the mall was put in the fridge.*)*

E. Others
This includes the remaining cases, such as:

(i) vocatives:
Pojď sem, Petře! (Come here, Peter!)
(i) particles and interjections that must be separated from the text by commas:
Haló, je tam někdo? (Hey, is anybody there?)
Zase jsi ve škole zlobil, že? (You were naughty at school again, weren't you?)

Table 1. *Estimated distribution of commas according to the presented typology. Total number of commas in the sample: 183*

Typology	# cases	frequency
A. comma precede the connective	94	51.4%
B. comma without the presence of the connective	49	26.8%
C. components of multiplied syntactic structure	31	16.9%
D. comma might but might not be inserted	8	4.4%
E. others	1	0.5%

3 Automatic Grammar Correction of Commas

The work presented in this paper was motivated by our experiments, where we initially compared the rule-based SET parser (Sec. 3.1) for automatic grammar correction of commas in Czech texts with the BERT-based method proposed in [23]. The BERT-based model was trained for slightly different task – restoring punctuation in recognized speech transcriptions. It is capable of inserting commas, full-stops and question marks. Based on the promising results, we trained a new model targeted only to the comma correction task (Sec. 3.2).

3.1 Rule-based approach

The SET system, firstly introduced in [14], was originally developed as a syntactic parser that matches patterns within text. Besides the main grammar for syntactic analysis, the

system also contains specialised sets of patterns that deal with subtasks of the automatic language checking (e. g. commas detection and correction, subject-predicate agreement, capitalization, ungrammatical structures such as zeugma, incorrect pronoun forms, etc.). These sets of patterns ("grammars") are used as the foundation of modules for the new online proofreader tool for the Czech language [8]. Before matching patterns, the text has to be tokenized and tagged. For tokenization purposes, we use the *unitok* tokenizer [22]. After thorough testing and comparison of the result [16], we decided to use two systems for morphological analysis: (1) the *Majka* analyzer [25] and *Desamb* tagger [24], (2) the *MorphoDita* system [21].

Based on the comma insertion rules, the SET parser produces pseudo-syntactic trees that indicate where a comma should or should not be inserted in the sentence structure (by hanging a particular word under the <c> or <n> node), as illustrated in Figure 1.

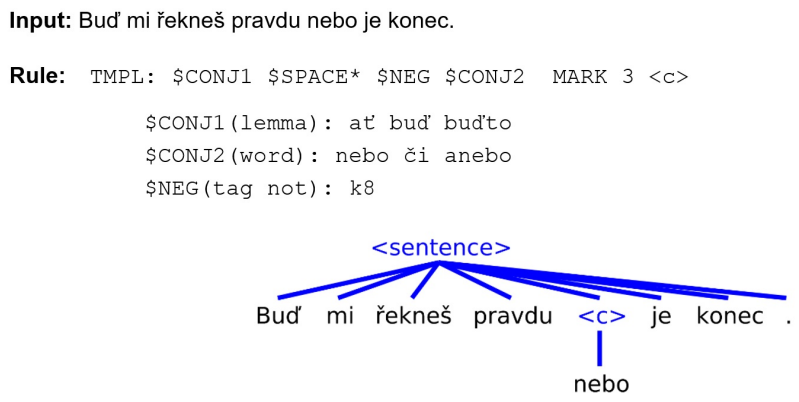

Fig. 1. One of the existing comma insertion rules, and its realization on a sample Czech sentence: *Buď mi řekneš pravdu nebo je konec* (missing comma before *nebo – Either you tell me the truth or it's over.*). In this case, the rule matches double-conjunction *buď – nebo* and if the conjunction *nebo* is not preceded by another conjunction (`$NEG(tag not): k8`), the analyzer inserts a comma before *nebo*.

The SET parser operates with a total of 1,400 rules. To formalize rules for type A (the comma precedes a connective) was the least complicated and had the highest recall. We can observe that Pareto's principle applies here - about 100 rules for subordinating conjunctions and relative pronouns cover 40% of all commas that should be inserted in a text. Next, we formalized rules for about ¼ of the commas of type B (the comma without the presence of the connective). These rules make use of the position of predicates or apply Wackernagel's law on the position of clitics in the sentence. The grammatical agreement of the case of components within a nominal phrase played a key role in formulating more than 600 rules for type C (components of multiplied

syntactic structure). For this category, we also tried to make use of the semantics of words using the *Word Sketch* function in the *Sketch Engine* software that provides a summary of a word's grammatical and collocational behaviour [11]. Nevertheless, the parser finds just over 10% of the commas of type C (that is little less than 2% of searched commas).The remaining categories D, and E are dealt with marginally - partly because the identification of the vocative based on morphological analysis is quite unreliable partly because the comma affects the meaning of the utterance.

The advantage of rule-based systems is the ability to correct or extend existing rules. For example, creating rules for the conjunction *ale* (lit. but) is a more complex task because *ale* can also be a particle or an interjection. The rules comprehensively evaluated in [13] inserted a comma before the conjunction *ale* with the precision of 94%. Now, after modifying these rules, the precision increased to 96.2% while maintaining the same recall. [1]

3.2 Transformer-based approach

Recent advances in using transformer models for NLP have inspired us to use Czech pre-trained RoBERTa model for automatic grammar correction of commas. Then this approach has been compared to the rule-based approach.

We used our own pre-trained model from a collection of web data processed in our web mining tool *[reference-hidden]*, Common Crawl data[2] and texts collected from the Czech part of Wikipedia. We followed all the training steps, mentioned in [15], for pre-training the Czech RoBERTa model. As suggested we used dynamic masking of the tokens. This strategy generates the masking pattern every time a sequence is fed to the model. Also, the pre-training procedure does not use Next Sentence Prediction loss in contrast with the BERT model [6]. For tokenizing the input sequence we used Byte-Pair Encoding (BPE), introduced in [20], with a subword vocabulary containing 50K units. This implementation of BPE uses bytes instead of Unicode characters as the base subword units.

We used the ADAM optimization with linear warmup up to learning rate$= 4 \cdot 10^{-4}$ for the first 24K steps followed by linear decay to 0. As the [15] suggested we pre-trained the model for 500K steps.

For this experiment, we proposed RoBERTA model extended by a few extra classification layers. An input sequence is preprocessed using the tokenizer and special tokens are added (as shown in the figure 2). Next, we use the Czech pre-trained RoBERTa model with output dimension d=768. RoBERTa last hidden states are transformed by four regular densely-connected neural network layers. Three of these layers use the element-wise ReLU activation function and the last layer uses the softmax activation function. The last layer output defines whether the comma should be placed right after the current token. The overall scheme of the proposed neural network architecture is depicted in the Fig.2.

As the training data set used for fine-tuning, we have used 10 GB of raw text extracted from the Czech CommonCrawl data set. Because the RoBERTa's output is related to

[1] You can try out the rule-based commas detection and correction at http://opravidlo.cz/

[2] https://commoncrawl.org/

input tokens (not words), we assigned the target label (”,” for comma, ”0” for no comma) to each token of the word, which is followed by a comma (as shown at the Fig.2). In the prediction phase, it is necessary to combine the predictions for the partial tokens into a word-level predictions using a per-word pooling. We use a simple average pooling algorithm to obtain the word-level predictions. As the model defines this experiment as a two classes classification per each token, we use standard categorical cross-entropy loss. For optimization, we use the standard Adam optimization algorithm.

We use epoch size equal to 10K sequences, the batch size equal to 45 and epoch size equal to 25, 50 and 75. During the fine-tuning, we use linear learning rate with values starting at 1^{-4}, ending at 1^{-5}. The maximum sequence length is set to 128.

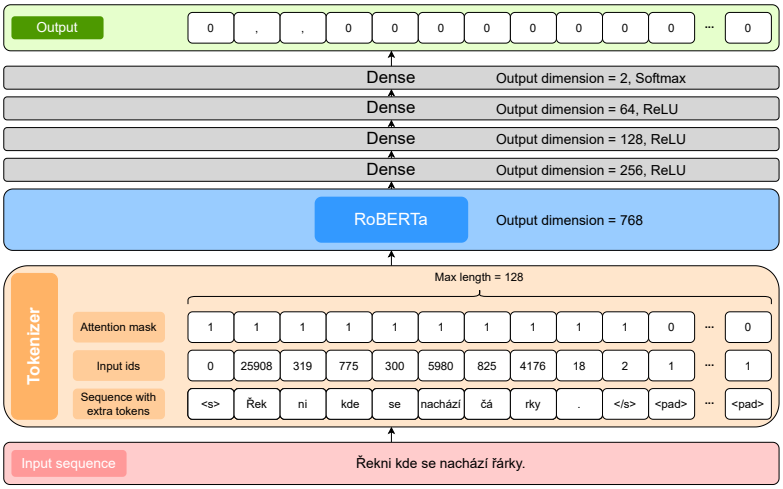

Fig. 2. Proposed model architecture. ”0” and ”,” in the output layer indicates where the commas should be placed. It is necessary to post-process the model output and align the commas at the end of the words composed of multiple tokens.

4 Evaluation data sets

The same data presented in [13] were used to evaluate and compare the methods described above. These texts were prepared specifically for the automatic insertion of commas, and since the data are exactly the same, it is also possible to compare the current results with testing done in the past. In total, 7 texts of different nature and styles were used: 3 sources from the Internet (blog texts, horoscopes and selected chapters from the *Internet Language Reference Book*), and 4 fiction books (2 Czech authors: one classic – K. Čapek, one contemporary – S. Monyová; 2 Czech translations of English authors: J. K. Rowling, N. Gaiman).

Blog texts and horoscopes were independently corrected by three proofreaders and the agreement rate among them was very high [13]. Selected chapters from the reference

book were written by linguists from the Academy of Sciences, and for the remaining texts, we assumed that they were proofread before being published. The size of the individual texts and the number of commas in them can be found in Tab. 2. Some may question whether this data is large enough for objective testing. On the other hand, the testing was done on real texts with their actual size and the writers themselves usually need to review texts of this size or smaller. Thus, we conclude that chosen data are appropriate for testing and evaluation.

Table 2. *Statistics of the test data [13].*

Testing set	# words	# commas
Selected blogs	20,883	1,805
Internet Language Reference Book	3,039	417
Horoscopes 2015	57,101	5,101
Karel Čapek – selected novels	46,489	5,498
Simona Monyová – Ženu ani květinou	33,112	3,156
J.K. Rowling – Harry Potter 1 (translation)	74,783	7,461
Neil Gaiman – The Graveyard Book (translation)	55,444	5,573
Overall	290,851	29,011

5 Experimental results

As explained in [13], probably the fairest evaluation method is the one where all commas are removed from a text and a selected tool inserts commas back into the text. Then, we can measure how many commas the tool inserted correctly and incorrectly in comparison with the original (reference) text. We use standard precision, recall and F1 score on detected commas.

The texts with commas removed, but still containing all the remaining punctuation, were first processed using the rule-based SET parser and the BERT-based punctuation filling method. The BERT-based baseline was implemented the same way as in [23] and was able to insert three punctuation symbols: comma, full stop and question mark. The results are shown in Tab. 3 and Tab. 4.

The overall F1 performance of the BERT-based model was above the rule-based approach. The performance gain is significantly higher on the first three datasets (blogs, the language reference book and horoscopes). The success of the trainable model motivated the work presented in this paper. We used the RoBERTA model and we fine-tuned it only on the comma insertion task. The results are summarized in Tab. 5. In comparison with the BERT baseline, the F1 scores improved for all types of evaluation texts.

If we compare the Roberta-based method with the rule-based baseline, the Precision scores are approximately the same. The difference is in the Recall scores, where the Roberta-based method provides consistently higher scores. The consistent increase in

Table 3. *Results: rule-based approach.*

	P [%]	R [%]	F1 [%]
Selected blogs	95.7	61.4	74.8
Internet Language Reference Book	92.7	42.5	58.2
Horoscopes 2015	96.9	71.3	82.2
Karel Čapek – selected novels	94.2	41.9	58.1
Simona Monyová – Ženu ani květinou	92.2	66.2	77.1
J.K. Rowling – Harry Potter 1 (translation)	94.7	61.7	74.7
Neil Gaiman – The Graveyard Book (translation)	96.3	56.5	71.2
Overall performance	95.1	58.8	72.7

Table 4. *Results: Transformer-based approach (baseline BERT [23]).*

	P [%]	R [%]	F1 [%]
Selected blogs	94.5	81.6	87.6
Internet Language Reference Book	93.7	67.4	78.4
Horoscopes 2015	94.6	88.4	91.4
Karel Čapek – selected novels	88.6	58.2	70.3
Simona Monyová – Ženu ani květinou	87.9	67.3	76.2
J.K. Rowling – Harry Potter 1 (translation)	89.3	68.1	77.2
Neil Gaiman – The Graveyard Book (translation)	91.2	63.7	75.0
Overall performance	90.9	69.7	78.9

Table 5. *Results: Transformer-based approach (Roberta).*

	P [%]	R [%]	F1 [%]
Selected blogs	95.5	88.3	91.8
Internet Language Reference Book	91.8	70.0	79.5
Horoscopes 2015	96.4	93.9	95.2
Karel Čapek – selected novels	95.3	88.9	92.0
Simona Monyová – Ženu ani květinou	95.8	93.1	94.4
J.K. Rowling – Harry Potter 1 (translation)	96.6	88.4	92.3
Neil Gaiman – The Graveyard Book (translation)	96.8	87.2	91.8
Overall performance	96.1	89.5	92.7

Recall (from 58.8% to 89.5%) also implies the increase in F1 score (from 72.7% to 92.7%).

6 Conclusion

In this paper, we compared the rule-based and the Transformer-based approaches to the task of automatic grammar correction of commas in the Czech language. The study was conducted using the production-ready representatives of each approach - the SET parser and Roberta-based model.

The experimental evaluation shows that training the Roberta-based model only for the comma insertion task provides better F1 performance, than using a generic BERT-based punctuation insertion model. A simple numerical comparison of the F1 scores shows consistently higher numbers for the Roberta-based model. On the other side, the Precision scores are comparable for the rule-based and Roberta-based approaches.

The great advantage of the rule-based SET parser is the explainability of the rules applied to the input sentence. The explainability of the Transformer models is unclear and almost impossible. The only way to interpret such models is to visualize the attention values. But this is complicated by the multi-layer and multi-head nature of the model.

RoBERTA model, for example, cannot insert a comma before the conjunction *a* (lit. and) followed by subordinating conjunction that starts a subordinate clause depending on the following main clause (*Pes čekal doma, a když pán přišel, radostně ho přivítal*). The model was unable to learn to deal with this type of syntactic construction because this type of comma does not usually appear in the training data from the Internet. We assume that the model could achieve even better results in the future if the training data will be without errors.

In future work, we would like to focus on the error analysis of the particular models. It would be helpful to know, what is the distribution of commas in our typology (A-E) among the erroneous predictions of the Roberta-based models. If there will be a common schema (for example lack of training data or consistent erroneous predictions), the rule-based approach can be used to extract the targeted training data and populate the rare cases to increase the overall performance. Another source of prediction errors in the Roberta-based model could be a mismatch between the training data (web text from CommonCrawl) and the evaluation data (especially the Internet Language Reference Book). Again, the rule-based model can be applied to pre-select the training data to better match the target dataset.

References

1. Pravidla českého pravopisu, 2. rozšířené vydání. Academia, Praha (1993)
2. Boháč, M., Rott, M., Kovář, V.: Text punctuation: An inter-annotator agreement study. In: Ekštein, K., Matoušek, V. (eds.) Text, Speech, and Dialogue. pp. 120–128. Springer International Publishing, Cham (2017)
3. Bryant, C., Felice, M., Andersen, Ø.E., Briscoe, T.: The BEA-2019 shared task on grammatical error correction. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 52–75. Association for Computational Linguistics, Florence, Italy (Aug 2019)

4. Cai, Y., Wang, D.: Question mark prediction by bert. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). pp. 363–367 (2019). https://doi.org/10.1109/APSIPAASC47483.2019.9023090
5. Chordia, V.: PunKtuator: A multilingual punctuation restoration system for spoken and written text. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. pp. 312–320. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-demos.37, https://aclanthology.org/2021.eacl-demos.37
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
7. Hlaváčková, D., Hrabalová, B., Machura, J., Masopustová, M., Mrkỳvka, V., Valíčková, M., Žižková, H.: New online proofreader for Czech. Slavonic Natural Language Processing in the 21st Century pp. 79–92 (2019)
8. Hlaváčková, D., Žižková, H., Dvořáková, K., Pravdová, M.: Developing online czech proofreader tool: Achievements, limitations and pitfalls. In: Bohemistyka, XXII, (1). pp. 122–134 (2022), https://doi.org/10.14746/bo.2022.1.7
9. Hlubík, P., Španěl, M., Boháč, M., Weingartová, L.: Inserting punctuation to asr output in a real-time production environment. In: Sojka, P., Kopeček, I., Pala, K., Horák, A. (eds.) Text, Speech, and Dialogue. pp. 418–425. Springer International Publishing, Cham (2020)
10. Karlík, P., Nekula, M., Pleskalová, J.e.: Nový encyklopedický slovník češtiny (2012–2020), https://www.czechency.org/
11. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The Sketch Engine: Ten Years on. Lexicography **1** (2014). https://doi.org/10.1007/s40607-014-0009-9
12. Klejch, O., Bell, P., Renals, S.: Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5700–5704 (2017). https://doi.org/10.1109/ICASSP.2017.7953248
13. Kovář, V., Machura, J., Zemková, K., Rott, M.: Evaluation and improvements in punctuation detection for czech. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Text, Speech, and Dialogue. pp. 287–294. Springer International Publishing, Cham (2016)
14. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for czech. In: Human Language Technology. Challenges for Computer Science and Linguistics. pp. 161–171. Springer, Berlin/Heidelberg (2011), https://doi.org/10.1007/978-3-642-20095-3_15
15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
16. Machura, J., Gerzová, H., Masopustová, M., Valícková, M.: Comparing majka and morphodita for automatic grammar checking. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN. pp. 3–14. Brno (2019)
17. Nunberg, G.: The Linguistics of Punctuation. CSLI lecture notes, Cambridge University Press (1990), https://books.google.cz/books?id=Sh-sruuKjJwC
18. Păiş, V., Tufiş, D.: Capitalization and punctuation restoration: a survey. Artificial Intelligence Review **55**(3), 1681–1722 (Mar 2022). https://doi.org/10.1007/s10462-021-10051-x,
19. Pravdová, M., Svobodová, I.: Akademická příručka českého jazyka. Academia, Praha (2019)
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

21. Straková, J., Straka, M., Hajič, J.: Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 13–18. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014). https://doi.org/10.3115/v1/P14-5003, https://aclanthology.org/P14-5003
22. Suchomel, V., Michelfeit, J., Pomikálek, J.: Text tokenisation using Unitok. In: Eight Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 71–75. Tribun EU, Brno (2014), https://nlp.fi.muni.cz/raslan/2014/14.pdf
23. Švec, J., Lehečka, J., Šmídl, L., Ircing, P.: Transformer-based automatic punctuation prediction and word casing reconstruction of the ASR output. In: Ekštein, K., Pártl, F., Konopík, M. (eds.) Text, Speech, and Dialogue. pp. 86–94. Springer International Publishing, Cham (2021)
24. Šmerk, P.: Unsupervised learning of rules for morphological disambiguation. Lecture Notes in Computer Science **3206** (2004), https://doi.org/10.1007/978-3-540-30120-2_27
25. Šmerk, P.: Fast morphological analysis of Czech. In: Proceedings of the RASLAN Workshop 2009. Masarykova univerzita, Brno (2009), https://nlp.fi.muni.cz/raslan/2009/papers/13.pdf