sciendo
DOI 10.2478/jazcas-2025-0024

TAILORED FINE-TUNING FOR COMMA INSERTION IN CZECH

JAKUB MACHURA[1] – HANA ŽIŽKOVÁ[2] – PATRIK STANO[3]
– TEREZA VRABCOVÁ[4] – DANA HLAVÁČKOVÁ[5] – ONDŘEJ TRNOVEC[6]
[1]Department Czech Language, Faculty of Arts, Masaryk University, Brno, Czech Republic (ORCID: 0000-0002-6623-3064)
[2]Department of Czech Language, Faculty of Arts, Masaryk University, Brno, Czech Republic (ORCID: 0000-0002-6483-6603)
[3]Department of Machine Learning and Data Processing, Faculty of Informatics, Masaryk University, Brno, Czech Republic (ORCID: 0009-0001-8339-6084)
[4]Department of Machine Learning and Data Processing, Faculty of Informatics, Masaryk University, Brno, Czech Republic (ORCID: 0009-0009-5674-3827)
[5]Department of Czech Language, Faculty of Arts, Masaryk University, Brno, Czech Republic (ORCID: 0000-0002-9918-0958)
[6]Department of Czech Language, Faculty of Arts, Masaryk University, Brno, Czech Republic (ORCID: 0009-0009-7756-9661)

MACHURA, Jakub – ŽIŽKOVÁ, Hana – STANO, Patrik – VRABCOVÁ, Tereza – HLAVÁČKOVÁ, Dana – TRNOVEC, Ondřej: Tailored Fine-tuning for Comma Insertion in Czech. Journal of Linguistics, 2025, Vol. 76, No 1, pp. 268 – 278.

Abstract: Transfer learning techniques, particularly the use of pre-trained Transformers, can be trained on vast amounts of text in a particular language and can be tailored to specific grammar correction tasks, such as automatic punctuation correction. The Czech pre-trained RoBERTa model demonstrates outstanding performance in this task (Machura et al. 2022); however, previous attempts to improve the model have so far led to a slight degradation (Machura et al. 2023). In this paper, we present a more targeted fine-tuning of this model, addressing linguistic phenomena that the base model overlooked. Additionally, we provide a comparison with other models trained on a more diverse dataset beyond just web texts.

Keywords: comma, Czech, Fine-tuning, Large Language Model (LLM)

1 INTRODUCTION

Punctuation, along with other graphical markers, plays a crucial role in ensuring the accurate comprehension of any text. The automatic insertion of sentence commas is typically addressed in two key tasks: (1) punctuation restoration in speech transcripts generated by automatic speech recognition (ASR), where reinstating punctuation significantly enhances readability, and (2) grammatical error correction in written texts, where commas may be either missing or redundant. In Czech, automatic punctuation correction is one of the most critical aspects of grammatical error correction, as the comma is not only the most frequently used punctuation mark (see Švec et al. 2021) but also a fundamental indicator of refined and structured writing.

For an extended period, the rule-based approach introduced by Kovář et al. (2016) demonstrated the highest precision in comma insertion. However, its recall did not exceed 60% of all commas. In recent years, advancements in machine learning have led to significant improvements in precision. The transformer-based approach proposed in (Machura et al. 2022) highlights the prevailing trend of training language models, analyzing their errors, and exploring potential refinements. This approach achieves precision comparable to or surpassing that of rule-based methods, while its recall exceeds 80%. A notable advantage of rule-based methods is their interpretability, as it is relatively straightforward to identify the specific rule responsible for a false positive. In contrast, neural network models function as black boxes, making it challenging not only to determine the cause of a particular error but also to implement targeted corrections.

This paper first extends previous experiments on re-training the RoBERTa model (see Section 2.1) and then introduces additional models fine-tuned on data from the SYN v9 corpus (Křen et al. 2021) available from the LINDAT/CLARIAH-CZ digital library. Finally, the study includes a comprehensive evaluation of all models using texts that have served as benchmarks for this task for nearly a decade (see Section 3).

2 COMMA INSERTION USING PRE-TRAINED MODELS

This section introduces various models pre-trained for the task of comma insertion. One set of experiments was conducted using a model trained and fine-tuned on web texts, many of which had not undergone any proofreading. Despite this, the results exceeded expectations, prompting the question of how models pre-trained or fine-tuned on higher-quality texts would perform.

In (Machura et al. 2022) the typology of the comma insertion place was comprehensively described. This allows 1) to specify the place (boundary) in the sentence structure where comma is inserted, 2) to analyze the type of commas that users of the language omit or overuse, or 3) to evaluate the results of language models that are pre-trained, namely for the task of inserting commas into text, and then subsequently improve these models.

Typology	Sample of newspaper articles with 183 sentence commas	
	# cases	frequency [%]
A. comma preceding 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. other types (vocative, particles, etc.)	1	0.5
decimal point	–	–

Tab. 1. Estimated general distribution of commas in Czech texts according to typology

2.1 RoBERTa – Training on web texts

Machura et al. (2023) examined the feasibility of re-training the RoBERTa language model, which had been pre-trained on a collection of web data, including Common Crawl[1] and texts from the Czech Wikipedia, and fine-tuned with a random data set from the Czech Common Crawl. The study focused specifically on improving RoBERTa's ability to detect commas in Czech vocatives by utilizing example sentences where the model had previously made errors. To this end, researchers extracted 170,000 sentences from the csTenTen17 corpus (Suchomel 2018) and employed two re-training strategies: (i) additional fine-tuning and (ii) an expanded training dataset, wherein the original large corpus was merged with a specialized corpus containing vocative phrases. While precision improved, recall declined significantly, likely due to overfitting to a specific comma type. The findings underscore the importance of training data distribution, highlighting the necessity of a broader dataset to preserve the model's overall functionality. The study ultimately demonstrates that while re-training RoBERTa is feasible, it requires careful structuring of the dataset to ensure balanced performance.

Following the vocative experiment, we have developed a fine-tuning dataset intended to enhance the overall performance of the RoBERTa base model for comma insertion – not solely for a specific type of comma. To identify RoBERTa's strengths and weaknesses, we conducted a thorough analysis on a dataset comprising 67,378 sentences and 87,379 commas extracted from news articles. The original texts (referred to as Gold) were presumed to be error-free following proofreading, although some errors did persist. By providing the model with these texts devoid of commas, it subsequently inserted 78,146 commas. The output (referred to as Test) was then compared with Gold using a script, revealing approximately 10,000 sentences where the model's comma placement diverged from the Gold standard.

Subsequently, we compiled a dataset of these sentences featuring mismatched commas (see the following Tab. 2), aligning each pair from Gold and Test side by side for annotation based on the aforementioned typology. Although the number of sentences identified by the script (10,000) slightly exceeded those annotated (8,890), this discrepancy likely arises from human annotation error and imperfect sentence separation by the script, particularly when segregating sentences in category A based on the connective following the comma.

The table below indicates that while categories A and C pose minimal challenges for the model, category B presents a moderate level of difficulty. In contrast, categories D and E are the most problematic. Comparing Tab. 1 with Tab. 2, the challenges associated with categories D and E are expected, given their lower relative distribution in the text, which results in a reduced amount of training data and consequently limits the model's ability to generalize effectively in these cases. Additionally, category

[1] https://commoncrawl.org/

D necessitates a deeper semantic and pragmatic understanding for accurate comma insertion. Based on these observations, we prioritized our fine-tuning efforts on more frequently occurring categories that offer greater potential for improvement.

Typology	Subcategory	# cases in subcategory	# cases in subcategory	Category frequency [%]
A. comma preceding the connective			1,777	19.99
B. comma without the presence of the connective	- asyndetic structures	1,336	2,726	30.66
	- right periphery of the embedded clause	1,102		
	- direct speech or quotation	288		
C. components of multiplied syntactic structure	- multiple sentence elements or enumeration	549	828	9.31
	- apposition	279		
D. comma might but might not be inserted	- non-restrictive attribute	169	1,530	17.21
	- multiple/sequential attribute	146		
	- comma changing the meaning	179		
	- constructions with *včetně*	107		
	- constructions with *jako*	106		
	- parentheses	358		
	- comma is not obligatory	465		
E. other types	- vocatives	129	355	3.99
	- particles and interjections	226		
Errors in Gold			855	9.62
Errors in Test			396	4.45
Cannot be determined			423	4.76
Total			8,890	

Tab. 2. Estimated distribution of the mismatched commas of the RoBERTa base model (Machura et al. 2022)

With this insight in mind, we compiled two datasets for fine-tuning RoBERTa. The first dataset, consisting of 1,313 sentences, was constructed using CQL queries on the internet corpus csTenTen2023 (Suchomel 2018) in Sketch Engine. Each sentence in this dataset was manually verified to ensure that it contained the correct type of comma as required by the CQL query. To identify sentences containing apposition, we utilized the syntactic function Apos in the syntactically annotated corpus SYN2020 (Křen et al. 2020) accessible via KonText (Machálek 2020). The second dataset, comprising 100,000 sentences, was entirely sourced from the SYN2020 corpus. This choice was motivated by the assumption that SYN2020 – composed solely of printed texts (fiction, non-fiction, newspapers, and magazines) – exhibits a higher linguistic standard compared to an internet corpus such as csTenTen2023, despite the absence of human verification for comma type accuracy. The CQL queries used to compile this larger dataset, along with the sentence counts for each query, are detailed in Tab. 3. Although the relative distribution of sentences

and the queries for the smaller dataset are largely consistent, minor differences exist due to the disparate morphological tagsets employed by each corpus manager. Again, two training strategies were used – (i) additional fine-tuning and (ii) an expanded training dataset, wherein the original large corpus was merged with a specialized corpus containing 1,313 or 100,000 sentences – yielding four model variants.

comma definition	CQL query	# cases
),	[lemma = "\)"][lemma = ","]	5,000
, "	[lemma =","][lemma = "\""]	7,000
",	[lemma = "\""] [lemma =","]	3,000
, a	[lemma = ","][lemma = "a"]	8,000
, aby	[lemma = ","][lemma = "aby"]	2,000
, ale	[lemma = ","][lemma = "ale"]	3,000
, co	[lemma = ","][lemma = "co"]	2,000
, či	[lemma = ","][lemma = "či"]	2,000
, jak	[lemma = ","][lemma = "jak"]	2,000
, jako	[lemma = ","][lemma = "jako"]	2,000
, kam	[lemma = ","][lemma = "kam"]	2,000
, kde	[lemma = ","][lemma = "kde"]	2,000
, když	[lemma = ","][lemma = "když"]	2,000
, (předložka) který	[lemma = ","][]{0,1}[lemma = "který"]	3,000
, nebo	[lemma = ","][lemma = "nebo"]	4,000
, než	[lemma = ","][lemma = "než"]	2,000
, protože	[lemma = ","][lemma = "protože"]	2,000
, že	[lemma = ","][lemma = "že"]	3,000
, (a/i/nebo) dokonce	[lemma = ","][lemma = "a" |lemma = "i" |lemma = "nebo"][lemma = "dokonce"]	2,000
buď – , anebo/nebo	[lemma = ","][lemma= "buď"][]*[word = ","][lemma = "anebo" | lemma = "nebo"] within <s/>	400
, ať – nebo/či	[word = ","][word = "ať"][]*[word = ","][word = "nebo" | word = "či"] within <s/>	1,000
asyndeton	[word = ","][tag !="J.*" & tag !="P[149EJKQ].*" & tag !="T.*"&tag !="R.*" & tag !="D.*" & word != "\""][word != "," & tag !="V.*"]{0,8}[tag = "V.*"] within <s/>	9,000
embedded clause	[word = ","][tag = "J.*"| tag="P[149EJKQ].*"| tag="D.*"][word != "," & word != "\."& tag != "V.*"]* [tag = "V.*"][word != "," & word != "\."& tag != "V.*"]*[word = ","][tag != "J.*" & tag !="P[149EJKQ].*" & tag !="D.*"] within <s/>	8,600
multiple sentence element (nouns)	1:[pos="N"][word=","] 2:[pos="N"] & 1.case = 2.case	3,000
multiple sentence element (adjectives)	1:[pos="A"][word=","] 2:[pos="A"] & 1.case = 2.case	2,000
multiple sentence element (verbs)	1:[pos="V"][word=","] 2:[pos="V"] & 1.tag = 2.tag	2,000
apposition	[afun = "Apos" & word=","]	6,000
constructions with *jako*	[lemma=","][lemma!="jako"]{0,1}[lemma="jako"]	3,000
, včetně	[lemma = ","][lemma = "včetně"]	3,000
particles and interjections	[tag="[IT].*"][lemma=","][lemma=","][tag="[IT].*"]	4,000

Tab. 3. List of CQL queries for compilation of 100,000 sentence dataset

2.2 Fine-tuning with SYN v9

To investigate the effectiveness of fine-tuning for automatic comma insertion in Czech text, we trained three different transformer-based models: RobeCzech-base (Straka et al. 2021), XLM-RoBERTa-large (Conneau et al. 2020), and mT5-large

(Xue et al. 2021). The RobeCzech-base and XLM-RoBERTa-large models were fine-tuned as token classification models, where the objective was to predict whether a given token should be followed by a comma. The mT5-large model was fine-tuned as a text-to-text model with the objective of adding commas to a text without any commas.

***Training Setup**:* Each model was trained using the SYN v9 dataset (Křen et al. 2021), available in the LINDAT repository, which was filtered to include only lines containing at least one comma. SYN v9 was chosen because the training of the RoBERTa baseline model was done on random texts from the internet and achieved quite good results, so the idea was to use texts that had mostly undergone some proofreading and might contain a wider variety of comma types. The dataset from SYN v9 was selected for its diverse curated content, as prior research (Machura et al. 2023) demonstrated that fine-tuning on an unfiltered Common Crawl dataset yielded significant results. However, even here, the comma type is random and may not match the frequency distribution of each comma type. Models were trained on datasets of 100,000, 300,000, and 500,000 lines from SYN v9, with experiments conducted using various numbers of training epochs. The best-performing hyperparameters for each model are listed below. Training and evaluation were performed on a single Nvidia A40 GPU, employing the AdamW optimizer and cross-entropy loss function.

	RobeCzech-base	XLM-RoBERTa-large	mT5-large
Dataset size	300k	300k	500k
Batch size	448	100	8
Learning rate	1e-5	1e-5	2e-5
Number of epochs	300	100	20

Tab. 4. The best hyperparameters for individual models

Preprocessing steps included tokenization using the respective model's tokenizer, as well as ensuring that quotation marks were tokenized as a separate token, and an optional transformation during evaluation where quotation marks were removed from the text. The impact of this transformation was analyzed in the evaluation phase (see Section 3).

2.3 Grammatical Error Correction (GEC)

In this experiment, we explore the application of the sequence-to-labels approach to grammatical error correction (GEC) for restoring missing commas in the text. This approach was inspired by the sequence labeling methods often used for the named entity recognition (NER) task (Kumar et al. 2023), as well as parts of the

GEC implementation of the grammarly/GeCToR architecture (Omelianchuk et al. 2020). Unlike in the more common sequence-to-sequence approach where the output is only the corrected input text, this approach returns both the corrected text and the labels showing where the changes have occurred, making it easier to interpret the model's decision.

We have prepared the training and evaluation datasets by introducing synthetic mistakes in the text, namely removing all commas from the text. Output of our preprocessing were pairs of documents:

- plain-text document (all commas were removed)
- label document where each word is tagged with a corresponding label:
  $KEEP: The word is correct and should not be changed.
  $MISSING_PUNCT_,: A comma should be inserted after this word.

Using the prepared training and evaluation datasets, we have fine-tuned a pre-trained RobeCzech-base model (Straka et al. 2021), tokenizing our datasets using the base model's tokenizer. To properly align the tokens with the reference word-level labels, the original word's label is duplicated across all corresponding tokens. During the fine-tuning process we evaluate the models' performance using the precision, recall, and F1-score for the $MISSING_PUNCT_, label class. At the end of the fine-tuning we evaluate the model with the highest F1-score during training on the test dataset. As the model predicts labels per token, during post-processing we convert the token-level predictions back into word-level labels, aggregating predictions for each word and selecting the predicted label with the highest frequency. If multiple labels have the same frequency, one is arbitrarily selected.

2.4 GPT-4o

For comparison, we also conducted an initial experiment in comma insertion using a generative language model GPT-4o-2024-08-06 (OpenAI 2024)[2]. Employing a temperature setting of 0.1 and a prompt instructing the model – "You are an expert in writing sentence commas in Czech and always respond in JSON format. Your task is to add missing commas to sentences" – the model demonstrated promising performance. A notable issue with this approach, however, was that the model occasionally modified the sentences beyond merely adding commas (e.g. altering or inserting words, correcting grammar), thereby complicating direct sentence comparisons. Modified sentences accounted for about 3%. This challenge could potentially be mitigated by refining the prompt or implementing a feedback loop to ensure that only commas are modified.

[2] https://chat.openai.com/

3 EXPERIMENTAL RESULTS

The dataset presented in Kovář et al. (2016) was utilized to evaluate and compare the methods described above. These texts were specifically designed for automatic comma insertion. As the dataset remains unchanged, the current results can be directly compared with previous evaluations. In total, seven texts of varying nature and style were used, as shown in Tab. 5.

Testing set	# words	# commas
Selected blogs	20,883	1,805
Internet Language Reference Book (ILRB)	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**

Tab. 5. Statistics of the test data for automatic comma insertion

The highest F1 score (93.1%) was achieved by the fine-tuned RobeCzech-base model when quotation marks were removed in preprocessing. The model outperformed the RoBERTa baseline model in terms of recall but exhibited lower precision. It is worth noting that in all RoBERTa baseline model experiments, post-processing was required for fiction texts, as the model consistently placed a comma after closing quotation marks in direct speech, despite the correct placement being before them. Overall, GPT-4o achieved the highest recall (92.0%); however, this came at the cost of precision, as it produced nearly 4,500 false positives (85.6%).

In the RoBERTa experiments (Section 3.1), an increase in training data consistently improved precision, reaching up to 98.2%; however, recall decreased significantly. The incorporation of additional datasets likely disrupted the frequency distribution of different comma types, leading the model to insert fewer commas with greater confidence. Notably, fine-tuning with the selected dataset, which was specifically designed to target phenomena ignored by the RoBERTa baseline model, yielded unexpected results, as all evaluation metrics declined.

Results of models from Section 3.2 – the RobeCzech-base and XLM-RoBERTa-large models showed improved performance when quotation marks were removed in preprocessing, while mT5-large achieved a better result with quotations included. A plausible hypothesis is that quotation marks can serve as useful syntactic cues for larger language models, aiding in the recognition of grammatical structures. For smaller models with more limited capacity, such as RobeCzech-base, they may act

as a source of noise or distraction. Despite being the smallest model, RobeCzech-base outperformed both XLM-RoBERTa-large and mT5. Its best performance surpasses a result reported in (Machura et al. 2022), while the other models failed to surpass this benchmark. The superior performance of RobeCzech-base suggests that a model specifically designed for Czech text may be more effective for this task than larger multilingual models. Further analysis could explore whether additional fine-tuning techniques or architectural modifications might enhance the performance of the larger models.

Section	Model	Precision [%]	Recall [%]	F1 [%]
3.1	RoBERTa baseline	95.9	89.3	92.5
	RoBERTa – Fine-tuning (1,313)	94.8	88.4	91.5
	RoBERTa – Fine-tuning (100,000)	95.7	87.5	91.4
	RoBERTa – Extended data (1,313)	97.8	79.3	87.6
	RoBERTa – Extended data (100,000)	**98.2**	75.8	85.5
3.2	RobeCzech-base	94.3	88.5	91.4
	RobeCzech-base “”*	94.5	91.7	**93.1**
	XLM-RoBERTa-large	94.6	85.9	90.0
	XLM-RoBERTa-large “”*	94.8	88.0	91.3
	mT5-large	95.1	85.9	90.3
	mT5-large “”*	95.6	84.1	89.5
3.3	Grammatical Error Correction	95.5	84.8	89.8
3.4	GPT-4o	85.6	**92.0**	88.7

“”* Evaluation without quotation marks

Tab. 6. Results of all mentioned models

4 CONCLUSION

The primary objective of this study was to develop a tailored dataset that incorporates linguistic phenomena overlooked by the RoBERTa baseline model. However, selecting the most frequently missing comma types to construct a retraining dataset did not lead to an improvement in the model's original performance.

The second objective was to compare models trained on web-based data – which, not having been proofread, often might contain false positives – with models trained on texts from the SYN v9 corpus, which are presumed to be of higher quality. The RobeCzech-base model fine-tuned on SYN v9 data outperformed the previous

RoBERTa model overall, but achieved a slightly lower precision. Further improvement could be achieved by filtering the SYN v9 dataset to be more representative of the natural frequency distribution of commas in Czech.

Additionally, an interesting comparison was made with GPT-4o and Grammatical Error Correction (GEC), both of which demonstrated comparable or superior performance in certain metrics. Nevertheless, their overall F1 scores remained relatively average.

In the next phase of this research, we will seek to identify the optimal composition of training data that encompasses all comma types in accordance with their natural frequency distribution, thereby maximizing recall. Simultaneously, the dataset must be balanced to achieve the highest possible precision, as the model must learn not only where to insert a comma—such as before a connective or other relevant token—but also where a comma should not be placed. For instance, while more than 4% of all commas in the SYN2020 corpus precede the conjunction *ale* 'but', in over one-quarter of all instances where *ale* 'but' appears, a comma is not required. Since neural networks function as a black box, we cannot determine with certainty whether this approach will produce the desired results. However, we believe that precisely constructing a balanced training dataset from SYN corpora could improve the functionality of the tested models.

ACKNOWLEDGEMENTS

The authors acknowledge that this work was supported by the OSCARS project, funded by the European Commission's Horizon Europe Research and Innovation program (grant agreement No. 101129751), led by the five Science Clusters: ENVRI, ESCAPE, LS RI, PaNOSC, and SSHOC.

References

Conneau, A. et al. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In: D. Juravsky et al. (eds.): Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 8440–8451.

Kovář, V. et al. (2016). Evaluation and improvements in punctuation detection for Czech. In: P. Sojka et al. (eds.): Text, Speech, and Dialogue. Springer International Publishing, pp. 287–294.

Křen, M. et al. (2020). SYN2020: A representative corpus of written Czech. Institute of the Czech National Corpus, Faculty of Arts, Charles University, Prague. Accessible at: http://www.korpus.cz.

Křen, M. et al. (2021). SYN v9: large corpus of written Czech, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of

Mathematics and Physics, Charles University. Accessible at: http://hdl.handle.net/11234/1-4635.

Kumar, P. et al. (2023). Transformer-Based Models for Named Entity Recognition: A Comparative Study. 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1–5. Accessible at: https://doi.org/10.1109/ICCCNT56998.2023.10308039.

Machálek, T. (2020): KonText: Advanced and Flexible Corpus Query Interface. In Proceedings of LREC 2020, pp. 7005–7010.

Machura, J. et al. (2022). Automatic Grammar Correction of Commas in Czech Written Texts: Comparative Study. In: P. Sojka et al. (eds): Text, Speech, and Dialogue. TSD 2022. Lecture Notes in Computer Science, Vol. 13502. Springer. Accessible at: https://doi.org/10.1007/978-3-031-16270-1_10.

Machura, J. et al. (2023). Is it possible to re-educate RoBERTa? Expert-driven machine learning for punctuation correction. In Slovko (October 18 – 20, 2023) Bratislava. Accessible at: https://dx.doi.org/10.2478/jazcas-2023-0052.

Omelianchuk, K. el al. (2020). GECToR – Grammatical Error Correction: Tag, Not Rewrite. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, Seattle, WA, USA → Online. Association for Computational Linguistics, pp. 163–170.

OpenAI. (2024). ChatGPT-4o. Accessible at: https://chat.openai.com.

Straka, M. et al. (2021). RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. In: K. Ekštein et al. (eds): Text, Speech, and Dialogue. TSD 2021. Lecture Notes in Computer Science, Vol. 12848. Springer, Cham. Accessible at: https://doi.org/10.1007/978-3-030-83527-9_17.

Suchomel, V. (2018). csTenTen17, a Recent Czech Web Corpus. In Twelveth Workshop on Recent Advances in Slavonic Natural Language Processing. Brno: Tribun EU, pp. 111–123.

Švec, J. et al. (2021). Transformer-based automatic punctuation prediction and word casing reconstruction of the ASR output. In: Ekštein, K. et al. (eds.): Text, Speech, and Dialogue, Springer International Publishing, pp. 86–94.

Xue, L. et al. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 483–498, Online. Association for Computational Linguistics.