Different Time, Different Language: Revisiting the Bias Against Non-Native Speakers in GPT Detectors

Adnan Al Ali[1] **and Jindřich Helcl**[2] **and Jindřich Libovický**[1]
[1] Charles University, Faculty of Mathematics and Physics
[2] University of Oslo, Language Technology Group
alali@ufal.mff.cuni.cz

Abstract

LLM-based assistants have been widely popularised after the release of ChatGPT. Concerns have been raised about their misuse in academia, given the difficulty of distinguishing between human-written and generated text. To combat this, automated techniques have been developed and shown to be effective, to some extent. However, prior work suggests that these methods often falsely flag essays from non-native speakers as generated, due to their low perplexity extracted from an LLM, which is supposedly a key feature of the detectors. We revisit these statements two years later, specifically in the Czech language setting. We show that the perplexity of texts from non-native speakers of Czech is *not* lower than that of native speakers. We further examine detectors from three separate families and find no systematic bias against non-native speakers. Finally, we demonstrate that contemporary detectors operate effectively without relying on perplexity.

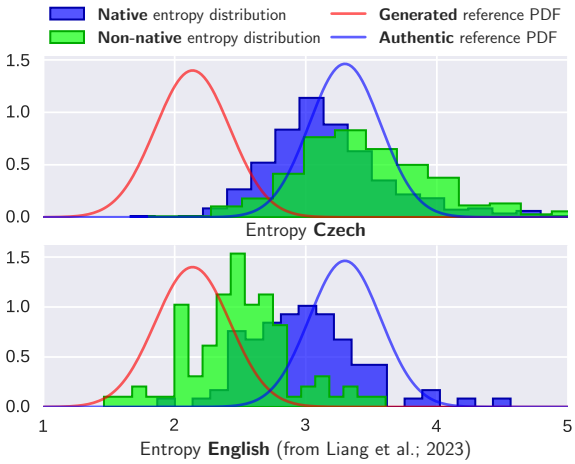

Figure 1: Distribution density of the entropy extracted from an LLM for essays from native vs. non-native speakers of Czech (top) and English (bottom). Unlike the English essays (Liang et al., 2023), we find that essays written by non-native speakers of Czech have higher entropies on average than those of their native peers. The reference PDF was computed on a Czech corpus.

1 Introduction

Following the release of LLM-based assistants – most notably ChatGPT, which was based on GPT-3 (Brown et al., 2020) and upgraded to GPT-4 (OpenAI et al., 2024a) and later versions – and their subsequent growth in popularity, concerns have emerged about the possible misuse of the service, particularly for plagiarism. This concern was largely raised in academic contexts (Susnjak and McIntosh, 2024).

Given the natural-sounding text generation, the distinction between human-written and generated text is challenging for humans (Ippolito et al., 2020; Milička et al., 2025). In contrast, machine-learning methods proved to be accurate to some extent (Wu et al., 2025). However, according to Liang et al. (2023), some of these methods are perplexity-based[1] and tend to be biased against non-native speakers of English, whose texts often have lower perplexities. As a result, the texts from non-native speakers might be falsely flagged as AI-generated.

In this article, we follow up on the work of Liang et al. (2023) in the Czech-speaking context and aim to answer three fundamental questions:

- **Q1** Is the perplexity of the texts from non-native speakers of Czech lower than that of the texts of their native peers?
- **Q2** Is there a bias against non-native speakers in Czech generated text detectors?
- **Q3** Is it possible to create generated text detectors without – explicitly or implicitly – relying on perplexity?

To answer **Q1**, we use an entropy-based analyser. Note that entropy and perplexity have a monotonic

[1] Perplexity in this context is measured with respect to an LLM; see Section 4 for definitions.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 277–291
March 26, 2026

exponential relationship. We measure the entropy distributions in all the inspected domains and compare the texts from non-native speakers with those of their native peers, as well as with texts from other domains.

For **Q2**, we examine a set of generated text detectors from the commonly used categories: (1) classical machine learning model using a bag-of-words text representation, (2) fine-tuned pre-trained RoBERTa-like model, and (3) closed–source commercial detector. We evaluate these models across multiple domains to assess their overall quality and performance on texts from native and non-native speakers.

Finally, to answer **Q3**, we inspect the correlations within one class (human-written or generated) between the outputs of the entropy-based analyser and the detectors, to assess whether they implicitly work with some internal representation of the entropy (or possibly explicitly in the case of the closed-source detector). We further examine how these correlations change across domains.

Our research reveals that the answers to the three proposed questions differ considerably from those presented in the seminal work of Liang et al. (2023), which has been reported in mainstream news outlets. The language setting proves to be a significant factor in considering the bias against non-native speakers.

The paper is structured as follows: Section 2 discusses the related previous work. Section 3 describes the creation of the datasets used in this work and their significance. In Section 4, we define the terms perplexity and entropy and use entropy analysis to address question **Q1**. In Section 5, we create and evaluate LLM detectors, addressing question **Q2**. Section 6 discusses how the entropy impacts the predictions of the detectors, addressing question **Q3**. Finally, we conclude our findings in Section 7.

2 Related Work

The concern of using LLM-generated text for academic misconduct has been raised since the release of ChatGPT. Susnjak and McIntosh (2024)[2] provided an early examination of the capabilities of ChatGPT in academic settings. The study underscored a concern that LLMs may pose a threat to academic integrity, especially in online examinations. The authors state LLM detectors as one of the prominent countermeasures to combat plagiarism.

2.1 Generated Text Detection

To our knowledge, no studies have been published on the creation/training of LLM detectors in Czech. Nonetheless, several multilingual evaluation benchmarks for LLM detectors contain Czech samples, such as *MULTITuDE* (Macko et al., 2023), which is based on a dataset of news articles (Varab and Schluter, 2021) and complemented with LLM-generated counterparts. The authors show that multilingual detectors fine-tuned on English, Spanish, and Russian samples can be zero-shot-transferred to Czech and maintain an F_1 score greater than 0.85. However, this dataset is limited to the news domain, which is a significant limitation, as detectors tend to be sensitive to domain changes (see below).

Various studies have been conducted on the automatic detection of generated text in English. Wu et al. (2025) provide a comprehensive survey on the problem. Below, we list three prominent approaches; however, this list is not exhaustive.

Classical machine learning methods using bag-of-words features in combination with SVMs, random forests, and logistic regression, among others, achieve performance comparable to more complex methods (Solaiman et al., 2019; Najjar et al., 2025) and serve as a solid baseline.

Logit-based methods use the raw outputs of a reference LLM. Solaiman et al. (2019) showed that the log-likelihood of a text under a model (the opposite value of entropy) is a useful feature but not satisfactory by itself, as this value differs across domains (Vasilatos et al., 2023). More complex logit-based methods have been successfully used for semi-automatic (Gehrmann et al., 2019) and automatic (Su et al., 2023; Mitchell et al., 2023) detection.

Fine-tuning a *pre-trained language model*, such as BERT (Devlin et al., 2019), has been a common approach (Solaiman et al., 2019; Fagni et al., 2021; Chen et al., 2023). However, such models have been shown to lack robustness when new domains or misspellings are introduced (Antoun et al., 2023), which is a key limitation.

2.2 Bias in GPT Detectors

Liang et al. (2023) examined how detectors of LLM work on text written by non-native speakers and found that the detectors systematically flag

[2]Preprint published in 2022.

Dataset	Description	#samples	Avg. #tokens	LLM?
SYNv9$^{\text{TRAIN}}$	Czech National Corpus + GPT-4o complement (train)	4766	511.57	Mix
SYNv9$^{\text{VAL}}$	Czech National Corpus + GPT-4o complement (val)	598	511.45	Mix
SYNv9$^{\text{VAL}}_{\text{4oMini}}$	Czech National Corpus 4o-mini complement	287	512.00	Yes
SYNv9$^{\text{VAL}}_{\text{Llama}}$	Czech National Corpus Llama complement	301	510.55	Yes
WIKI	Wikipedia crawl + GPT-4o complement	422	512.00	Mix
NEWS	News crawl + GPT-4o complement	762	511.94	Mix
NONNATIVE	Essays from non-native speakers	450	342.16	No
NONNATIVEC1	Essays from proficient non-native speakers	29	512.00	No
NATYOUTH	Essays from native speakers (children)	450	331.51	No
NATADV	Essays from native speakers (age 16–18)	29	496.59	No
ABS2020	Pre-GPT theses abstracts	1655	283.29	No
ABSNEW	Post-GPT theses abstracts	2050	298.51	Unk

Table 1: Overview of the datasets, their sample count, average number of `unitok` tokens per sample (after truncation to max. 512 tokens), and their LLM-generated status. While it is unclear whether (and to what extent) **ABSNEW** is generated, the metrics in this article assume it is human-written for simplicity.

the texts written by them as generated. The authors evaluated seven 'widely used' generated text detectors on two groups of documents: (1) Test of English as a Foreign Language (TOEFL) essays written by Chinese students (91 documents), and (2) Hewlett Foundation's ASAP dataset (Hamner et al., 2012), containing US eight-graders' essays (88 documents).

The study found that, in most cases, the evaluated detectors correctly labelled the essays from US students as human-written, with the mean False Positive Rate (FPR) being 5.1%. In contrast, the detectors often misclassified the TOEFL essays written by Chinese students as GPT-generated, with the mean FPR of 61.3%. Furthermore, all seven detectors unanimously flagged 19.8% of the TOEFL essays as AI-generated. Those essays have been shown to have low perplexities.

The authors further proceed to present a claim that *'most GPT detectors use text perplexity to detect AI-generated text'*. In our study, we follow up on the presented claims and test whether the texts from non-native speakers of Czech have smaller perplexities and whether we can create a classifier that does not rely on perplexity. Our findings in the Czech setting differ significantly from those in the prior study.

3 Datasets

Training and evaluation of detectors of LLMs requires carefully curated datasets of clear origin (human or LLM). For training, a reasonably large corpus (comprising millions of tokens) of high-quality text is required (both human-written and generated). Evaluation data must encompass diverse domains extending beyond the training data. Importantly for our purposes, the evaluation data must also contain texts from non-native speakers and comparable texts from native speakers. Table 1 contains the overview of our datasets.

3.1 Conteporary Czech Corpus

For training, we exclusively use SYNv9 (Křen et al., 2021), which is the most comprehensive collection of contemporary (synchronic) Czech corpora consisting of news/magazines (predominantly), non-fiction, and fiction domains. We randomly sampled 7460 texts published between the years 2009 and 2019. We truncated the texts to 2000 pre-annotated tokens. This dataset of authentic documents is referred to as **SYNv9$_{\text{AUTH}}$**.

We complement the authentic data with an LLM-generated counterpart. To match the structure and vocabulary of the authentic corpus, we used the prompt that contained a short sample of the texts from **SYNv9$_{\text{AUTH}}$**, resulting in one generation prompt for each text (see Appendix A for details).

We produced generated samples using various LLMs. As the primary source of generated text, we use GPT-4o[3] (OpenAI et al., 2024b), with the temperature 0.7 and number of tokens limited to 1024. We generated the data, discarded the files smaller than 2 kB, and split them into two subsets: **SYNv9$^{\text{TRAIN}}_{\text{GPT4o}}$** and **SYNv9$^{\text{VAL}}_{\text{GPT4o}}$**. For training, we paired the **SYNv9$^{\text{TRAIN}}_{\text{GPT4o}}$** samples with the authentic samples that were used to generate them, resulting in the training set **SYNv9$^{\text{TRAIN}}$**. Analogously, we created **SYNv9$^{\text{VAL}}$**.

To include more models for validation, we

[3]Version `gpt-4o-2024-05-13`.

created $\textbf{SYNv9}^{\text{VAL}}_{\text{4oMINI}}$ using GPT-4o-mini[4] and $\textbf{SYNv9}^{\text{VAL}}_{\text{LLAMA}}$ using Llama 3.1 405B[5] (Grattafiori et al., 2024) analogously.

3.2 Wikipedia and Online News Crawl

To include more domains, we added Wikipedia and online news articles for evaluation. The $\textbf{WIKI}_{\text{AUTH}}$ dataset was created by crawling Wikipedia using `pywikibot`.[6] Articles were chosen randomly, retrieved and parsed using `mwparserfromhell`.[7] Articles that contained fewer than 1000 space characters were discarded. The generated complement was created using GPT-4o, prompted to write a Wikipedia article on a given topic from $\textbf{WIKI}_{\text{AUTH}}$. After filtering, 211 articles were created ($\textbf{WIKI}_{\text{GPT4O}}$) and paired with their authentic counterparts, creating **WIKI**.

We sampled the $\textbf{NEWS}_{\text{AUTH}}$ dataset randomly from the web news crawl 2021 (Kocmi et al., 2022).[8] Again, we generated a complement using GPT-4o with an analogous prompt, creating 381 generated articles ($\textbf{NEWS}_{\text{GPT4O}}$) and paired them with their counterparts, resulting in **NEWS**.

3.3 Non-Native and Native Youth Works

As a crucial dataset for determining the performance of our classifiers on text by non-native speakers, we utilised the AKCES 3 corpus (Šebesta et al., 2012), a corpus of essays written by non-native students of the Czech language. To filter out texts with too frequent mistakes, we only included speakers who had studied Czech for at least 24 months at the time of writing. The dataset is referred to as **NONNATIVE**.

In an effort to better reproduce the work of Liang et al. (2023), we created **NONNATIVEC1**, a dataset of advanced non-native speakers, sourced from an extended version of AKCES 3 (Náplava and Straka, 2019), from speakers with language proficiency labelled as proficient.[9] Finally, we filtered the 118 samples, which we found to still contain frequent errors, to be at least 2 kB in size, yielding 29 samples, predominantly from Slavic authors.

We state the distributions of the L1 languages of the non-native datasets in Appendix B.

To roughly match the domain of **NONNATIVE**, we utilised the AKCES 1 corpus (Šebesta et al., 2016), a collection of essays written by native Czech speakers at primary (from 5th grade) and secondary schools. Furthermore, we attempted to match the distribution of the file size of the native texts to the **NONNATIVE** dataset by selecting the most similar (in file size) text from AKCES 1 for each text in the **NONNATIVE** dataset. We denote the resulting subset of AKCES 1 by **NATYOUTH**.

To match the advanced non-native texts from **NONNATIVEC1**, we randomly selected 29 texts from AKCES 1 with the age category labelled as 'over 15 years' – i.e. 16–18 years. We denote this dataset by **NATADV**.

3.4 Academic Abstracts

To evaluate the models on academic texts, we created a corpus of theses abstracts, crawled from the Charles University Digital Repository,[10] published between 2020 and 2021 – i.e., before the introduction of ChatGPT. We denote this dataset by **ABS2020**.

For comparison, we also included the abstracts written between 2023 and 2025 – after ChatGPT became widely popular – yielding **ABSNEW**.

4 Entropy and Perplexity

To address question **Q1** – i.e. whether non-native speakers of Czech tend to produce texts that LLMs rate with smaller perplexity compared to their native peers – we redefine the task using entropy and analyse the datasets. In this context, perplexity is defined as the *'exponential average negative log-likelihood of a token sequence given a specific language model'* (Jiang et al., 2024).

Omitting the exponentiation, we can define entropy as the per-token average negative log-likelihood of a document given an LLM:

$$-\frac{1}{N}\sum_{i=1}^{N}\log P(d_i \mid d_1, \ldots, d_{i-1}) \qquad (1)$$

where $(d_1, \ldots, d_N) = \mathbf{d}$ is a token sequence.

We chose to work with entropy rather than perplexity, as it roughly follows a Gaussian distribution (as shown in Figure 2). For our purposes, the two metrics are interchangeable, as they have a

[4] Version `gpt-4o-mini-2024-07-18`; https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence

[5] Ollama model `llama3.1:405b-instruct-q5_K_S`

[6] https://www.mediawiki.org/wiki/Manual:Pywikibot

[7] https://github.com/earwig/mwparserfromhell

[8] Although the news domain is contained in **SYNv9**, the structure of online news articles may likely be different from the printed ones.

[9] C1 or C2 under the CEFR.

[10] https://dspace.cuni.cz; the largest database of theses in Czech.

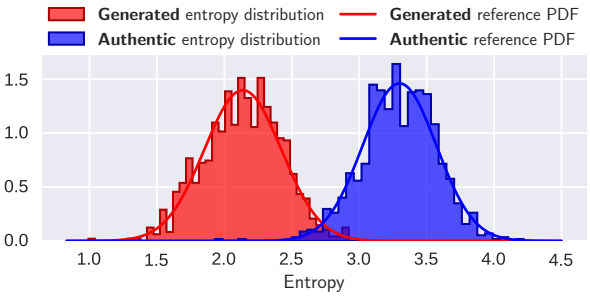

Figure 2: Distribution density of the entropy for generated and authentic samples from **SYNv9**$^{\text{TRAIN}}$, together with their fitted Gaussian PDF.

monotonically increasing relationship, preserving the ordering.

4.1 Entropy Analysis

For our reference model, we chose Llama 3.2 1B base (Grattafiori et al., 2024), unlike previous work (Liang et al., 2023; Jiang et al., 2024), which used GPT-2 (Radford et al., 2019). This is because GPT-2 performs poorly on Czech (Hájek and Horák, 2024) and is somewhat outdated.[11] Nonetheless, we found that our Llama-based entropy analyser produces comparable results to those reported by Liang et al. (2023) when applied to their dataset (see Table 2 bottom and Figure 1).

The number of samples for each dataset was clipped to 1000, and the subset was selected at random. We truncated each sample to 512 tokens and disregarded the predictions on the first 50 tokens to provide sufficient context, for better stability. Let M denote the length of the truncated document, then our modified entropy formula is as follows:

$$-\frac{1}{M-50}\sum_{i=51}^{M}\log P(d_i \mid d_1,\ldots,d_{i-1}) \qquad (2)$$

We display the distribution density together with the fitted Gaussian PDFs for **SYNv9**$^{\text{TRAIN}}$ in Figure 2 and the distributions for the remaining datasets in Appendix F. In most cases, the generated documents have smaller entropies than authentic, although some overlap is present.

4.2 Results

Table 2 reveals a key finding (illustrated in Figure 1): non-native (**NonNative**) speakers produce texts with greater entropy than native (**NatYouth**) speakers do ($p < 10^{-14}$), opposed to the findings of Liang et al. (2023). On average, this is also the case for the advanced essays corpora (**NonNativeC1** vs. **NatAdv**), although the difference is not significant ($p > 0.19$).

Dataset	Generated		Natural	
	Mean	SD	Mean	SD
SYNv9$^{\text{TRAIN}}$	2.14	0.29	3.30	0.27
SYNv9$^{\text{VAL}}$	2.13	0.29	3.30	0.27
SYNv9$^{\text{VAL}}_{\text{4oMini}}$	2.09	0.26	–	–
SYNv9$^{\text{VAL}}_{\text{Llama}}$	1.97	0.36	–	–
Wiki	1.67	0.20	2.4	0.30
News	1.89	0.18	2.82	0.36
NonNative	–	–	3.48	0.57
NonNativeC1	–	–	2.97	0.36
NatYouth	–	–	3.19	0.49
NatAdv	–	–	2.85	0.22
Abs2020	–	–	2.34	0.36
AbsNew	–	–	2.33	0.35
TOEFL-91 (en)	–	–	2.5	0.39
Hewlett (en)	–	–	2.99	0.44

Table 2: The mean and standard deviation (SD) of the entropy for each dataset. The last two rows display the entropy of the English datasets used by Liang et al. (2023): **TOEFL-91** (non-native) and **Hewlett** (native).

Another finding is that the entropy of the essays gets smaller as the students get more advanced (**NonNative** vs. **NonNativeC1**; $p < 10^{-6}$). We investigated this on a token-level and concluded that introducing grammar errors indeed lowers the probability of the tokens of a misspelt word, increasing the entropy. Appendix C contains an illustrative example of this phenomenon.

Non-native speakers, therefore, tend to produce two types of features in the text, which have opposite effects on entropy: limited vocabulary (decreasing the entropy) and grammar errors (increasing the entropy). While Liang et al. (2023) found that the prior is more prominent in English, we found that the latter is more prominent in Czech, which has more complex morphology.

We further observe that: (1) The entropy distribution does not differ significantly between **SYNv9**$^{\text{VAL}}_{\text{4oMini}}$ and **SYNv9**$_{\text{GPT4o}}$ ($p > 0.42$). (2) On average, **SYNv9**$^{\text{VAL}}_{\text{Llama}}$ has slightly smaller entropy compared to its GPT counterparts, likely caused by the reference model belonging to the same family. (3) The entropy differs across domains; e.g., both **Wiki** and **News** have smaller entropies compared to the SYNv9 datasets – likely because their domains were included in pre-training data. (4) The entropy does not differ significantly between **Abs2020** and **AbsNew** ($p > 0.1$).

[11] Another option would be GPT-OSS (OpenAI et al., 2025). However, this model is not published in its base version, possibly making the entropy calculation less reliable.

5 Generated Text Detection in Czech

In order to examine question **Q2** – whether there is a bias against non-native speakers of Czech in generated text detectors – we work with three classes of detectors: (1) a naïve Bayes (NB) detector with TF-IDF features, as a baseline (2) a fine-tuned RoBERTa-like detector, and (3) a commercial multilingual[12] closed-source detector.

5.1 Naïve Bayes Detector

As a typical approach (Kibriya et al., 2005), we chose the combination of TF-IDF features and multinomial NB, using the `unitok` tokeniser (Suchomel et al., 2014). We trained the detector on the **SYNv9**$^{\text{TRAIN}}$ dataset with lowercasing as pre-processing and truncation to 512 tokens. We discuss the training details in Appendix D.

Dataset	Acc	FPR	FNR	Unk
SYNv9$^{\text{TRAIN}}$	99.1	0.8	1.0	13.1
SYNv9$^{\text{VAL}}$	99.2	1.0	0.7	15.4
SYNv9$^{\text{VAL}}_{\text{4oMini}}$	99.0	–	1.1	9.4
SYNv9$^{\text{VAL}}_{\text{Llama}}$	73.1	–	26.9	13.8
Wiki	88.9	9.4	13.3	25.1
News	98.2	3.4	0.3	15.0
NonNative	91.3	8.7	–	22.2
NonNativeC1	75.9	24.1	–	14.8
NatYouth	93.8	6.2	–	17.8
NatAdv	75.9	24.1	–	15.8
Abs2020	19.8	80.2	–	17.0
AbsNew	17.9	82.1	–	17.2

Table 3: Evaluation results of the **TF-IDF NB** classifier. Values are shown as per cent (%). The 'Unk' column corresponds to the ratio of out-of-vocabulary tokens in the dataset.

Results. The results in Table 3 show that (1) There is no significant difference in performance on the native and non-native datasets ($p > 0.23$ for **NonNative** vs. **NatYouth**; $p > 0.81$ for **NonNativeC1** vs. **NatAdv**). (2) The NB detector achieves a near-perfect performance on the in-domain validation data (**SYNv9**$^{\text{VAL}}$) but deteriorates considerably on different domains. (3) The detector is not robust to changing the source model, performing considerably worse on **SYNv9**$^{\text{VAL}}_{\text{Llama}}$.

5.2 RobeCzech Detector

We chose RobeCzech (Straka et al., 2021) as the base model for our RoBERTa-like (Liu et al., 2019) detector, as it is the best-performing Czech monolingual model of its type. The architecture consisted of the base model and a classification head attached to the final representation of the special `[CLS]` token. The context length of RobeCzech is 512 tokens, which leads to truncation.[13] We discuss the details about the architecture and the training procedure in Appendix E.

Results. The results in Table 4 (left) show that, similarly to the NB detector, the RobeCzech detector achieves a near-perfect performance on the training domain but lacks robustness on others. This is consistent with the findings of Antoun et al. (2023). Regarding the potential bias, the performance on the datasets from native speakers was considerably higher. However, upon inspection, we discovered that the relatively good performance on **NatYouth** was caused by the presence of the *non-breaking space* character, and its replacement with a regular space led to a decrease in accuracy from 71.6% to 42.4%, falling short of the **NonNative** dataset.

The role of the non-breaking space is somewhat paradoxical, as it is not a part of the training dataset (or any dataset other than **NatYouth**). We hypothesise that the model has developed some generalised rule associating rare tokens with the *'human-written'* label. In an attempt to mitigate this, we introduced random data augmentation (RDA) using random noise. This involved adding random sequences of Unicode characters and randomly mutating the whitespace characters to sequences of other whitespace characters. Appendix E.3 describes the details of the augmentation.

Table 4 (right) contains the results after applying the RDA pipeline during training and inference. While the performance generally improved, it still did not consistently surpass the 50% random baseline. Moreover, the detector performed inconsistently on the native vs. non-native comparison, performing better on **NonNative** than **NatYouth** ($p < 0.03$; $\Delta\text{FPR} = 5.1\%$) but worse on **NonNativeC1** than **NatAdv** on average, although not significantly ($p > 0.98$).

5.3 Commercial Detector

Finally, to provide more realistic detection results, we included a commercial, closed-source model for comparison. After an informal survey of the avail-

[12] No functional monolingual detector for Czech exists, as of writing this article.

[13] We truncate the samples in other detectors to 512 tokens too, to provide comparable settings.

Dataset	No Augmentation			Random Augmentation		
	Acc	FPR	FNR	Acc	FPR	FNR
SYNv9$^{\text{TRAIN}}$	99.4	0.4	0.9	98.9±0.0	0.7±0.0	1.4±0.1
SYNv9$^{\text{VAL}}$	99.3	0.3	1.0	99.0±0.1	0.9±0.3	1.0±0.0
SYNv9$^{\text{VAL}}_{\text{4oMini}}$	99.7	–	0.4	99.7±0.0	–	0.4±0.0
SYNv9$^{\text{VAL}}_{\text{Llama}}$	92.7	–	7.3	92.7±1.4	–	12.6±1.4
Wiki	86.7	15.2	11.4	86.7±1.0	10.7±0.7	15.9±1.4
News	86.0	27.8	0.3	92.7±0.4	14.0±0.7	0.6±0.2
NonNative	52.4	47.6	–	67.4±0.9	32.6±0.9	–
NonNativeC1	10.3	89.7	–	24.1±2.4	75.9±2.4	–
NatYouth	71.6	28.4	–	62.3±0.7	37.7±0.7	–
NatAdv	37.9	62.1	–	33.8±2.9	66.2±2.9	–
Abs2020	33.8	66.2	–	65.1±0.5	35.0±0.5	–
AbsNew	32.4	67.6	–	62.1±0.3	37.9±0.3	–

Table 4: Evaluation results of the **RobeCzech** classifier with no augmentation (left) and RDA (right) applied on training and inference, together with the 5-trial evaluation standard deviation (the model was only trained once). Values are shown as per cent (%).

able options, we found that *Plagramme*[14] performs well on the tested documents. The tool operates at the sentence level, returning the classification probability for each sentence in the document. To obtain the same format as our previous detectors, we compute the average of the probabilities (each sentence with the same weight).

Dataset	Acc	FPR	FNR
SYNv9$^{\text{VAL}}$	97.5	1.0	4.0
SYNv9$^{\text{VAL}}_{\text{4oMini}}$	99.0	–	1.0
SYNv9$^{\text{VAL}}_{\text{Llama}}$	65.0	–	35.0
Wiki	93.0	2.0	12.0
News	96.5	6.0	1.0
NonNative	98.0	2.0	–
NonNativeC1	100.0	0.0	–
NatYouth	99.0	1.0	–
NatAdv	96.6	3.5	–
Abs2020	96.0	4.0	–
AbsNew	89.0	11.0	–
TOEFL-91 (en)	76.9	23.1	–
Hewlett (en)	100.0	0.0	–

Table 5: Evaluation results of the **Plagramme** detector. Values are shown as per cent (%). The last two rows display the performance on the English datasets used by Liang et al. (2023): **TOEFL-91** (non-native) and **Hewlett** (native).

Due to API constraints, we limited the size of each dataset to 100 randomly selected documents, truncated each document to 512 words, and normalised the whitespace to a single space character.

Results. The results in Table 5 show considerably better performance than the classifiers we previously created and presented, indicating that our detectors fail to achieve the SoTA performance. The detector struggled the most on the SYNv9$^{\text{VAL}}_{\text{Llama}}$ dataset, suggesting that it was not trained on Llama-generated (Grattafiori et al., 2024) documents and does not generalise well across the models. The difference in performance on the native vs. non-native datasets was not significant and inconsistent: performing better on **NatYouth** than on **NonNative** ($p > 0.11$) but worse on **NatAdv** than on **NonNativeC1** ($p > 0.15$). Finally, the detector flagged 11% of the post-GPT abstracts (**AbsNew**) as generated, which may reflect reality.

Results on the English datasets. As a side experiment, we leveraged the detector's multilinguality to evaluate it on the datasets from Liang et al. (2023). While the accuracy measured on the non-native dataset was smaller than the native dataset by a non-trivial margin ($\Delta\text{FPR} = 23.1\%$), notable progress has been made since 2023: the FPR improved from the reported 61.3% (mean) or 48% (best detector) to our observed 23.1%. Moreover, the correlation between text entropy and detector output was negligible and slightly positive[15] ($0 < \rho < 0.04$), suggesting that another factor caused the drop in performance.

5.4 Discussion

The results presented in this section demonstrate that creating a robust detector of generated text is a feasible, yet non-trivial task. As an answer to question **Q2**, we find that *none of the detectors exhibited a systematic bias against non-native speakers of Czech* when compared with their native peers. We further find that the bias against

[14] https://www.plagramme.com/services/ai

[15] Contrary to our expectation of a negative coefficient, as low-entropy samples supposedly receive positive labels.

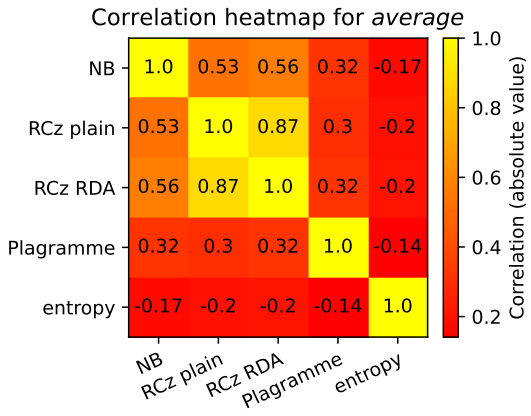

Figure 3: Per-dataset **average** correlation heatmap for the compared models. Key: NB: naïve Bayes detector; RCz plain: RobeCzech detector with no augmentation; RCz RDA: RobeCzech detector with random data augmentation.

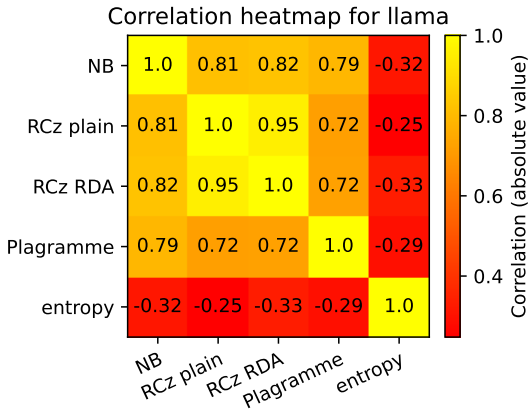

Figure 4: Correlation heatmap for the compared models, for the $\mathbf{SYNv9}^{\mathbf{VAL}}_{\mathbf{LLAMA}}$ dataset. Key: see Figure 3.

non-native speakers of English, as measured on the dataset from Liang et al. (2023), is considerably less pronounced in the contemporary detector than originally reported.

6 Correlation Analysis

Finally, we address question **Q3**, whether the presented detectors rely on perplexity, or entropy (in our case). While our custom detectors do not have explicit access to the entropy, they may still have some internal representation of it. We test the relationship by calculating the in-class Pearson correlation coefficients between the entropy (as defined in Equation 2) and the outputs of the models.

For completeness, we computed the correlation between all pairs of detectors presented in the article. **AbsNew** was excluded to ensure that we do not compute the correlation on a potentially mixed-class dataset. We show the heatmaps for all datasets in Appendix G.

6.1 Results

Figure 3 shows that the per-dataset mean correlation between the entropy and the outputs of all models is negative, as expected (low entropy is a feature of documents with a high positive classification probability), but very weak ($|\rho| \leq 0.2$). Moreover, the correlation between Plagramme and our custom detectors is also quite low, suggesting that they work on a different principle.

Interestingly, all of the correlations were stronger in the $\mathbf{SYNv9}^{\mathbf{VAL}}_{\mathbf{LLAMA}}$ dataset, which we show in Figure 4. This may suggest that the RobeCzech and Plagramme classifiers might work with some more-complex GPT-specific patterns but fall back to lexical features (typical for the NB classifier) when those patterns are not present.

7 Conclusion

We provide a comprehensive follow-up to the work of Liang et al. (2023), who claim that GPT detectors are biased against non-native speakers. Our work differs from the previous in two ways: time setting (working with the models and detectors available in 2025 rather than 2023) and language setting (working with Czech rather than English). Under these changes, we draw considerably different conclusions. We inspect the claims using a diverse set of datasets covering, among others, essays from non-native speakers and comparable essays from native speakers.

First, we develop a method for measuring the entropy of a text in a stable way. We apply this method to our datasets and find that *essays by non-native speakers have entropy no lower than essays by native speakers*. In fact, it is slightly greater, which is likely due to frequent grammatical and spelling errors that contribute to entropy. The errors may be more prevalent in Czech than in English because of its complex morphology.

Next, we attempt to train our custom detectors of LLM-generated text from two families – TF-IDF naïve Bayes and RoBERTa-like (Liu et al., 2019) models – and find that training a detector robust to different domains is a non-trivial task. Nonetheless, our *detectors did not consistently exhibit any biases*. Moreover, we demonstrate that commercial detectors achieve satisfactory results across all domains without exhibiting bias either.

Finally, we analyse whether the presented detectors rely (explicitly or implicitly) on the entropy using a correlation analysis. Our findings show that the correlation between the models' outputs and the entropy is very weak ($|\rho| \leq 0.2$), suggesting that the *models do not largely depend on the entropy*.

We conclude that the *bias in GPT detectors is language dependent* and likely sensitive to the morphology of the specific language. Future work may conduct similar experiments on more languages to better understand the relationship. Moreover, we conclude that the technologies for detecting generated text have improved considerably since 2023, yielding more satisfactory results.

Limitations

The sourcing of the datasets was subject to several limitations. Access to large corpora of proficient non-native speakers of Czech is limited. As a result, we used a reasonably sized dataset of essays from moderately proficient speakers (450 documents) that contained frequent errors, and a small dataset of essays from proficient speakers (29 documents). Furthermore, we only used a limited number of source LLMs for our documents, despite the existence of different families and more advanced models.

We encountered limitations during the creation of the detectors as well. Notably, we were unable to reach the SoTA performance with our custom detectors. We partially addressed this by including a robust, commercial detector. However, given its proprietary nature, the analysis was limited to 'black-box' observations only.

Acknowledgments

We thank Zdeněk Kasner for his valuable feedback and *Plagramme* for providing access to their otherwise non-public API.

Adnan was supported by the HumanAId project `CZ.02.01.01/00/23_025/0008691` of the Czech Ministry of Education. Jindřich H. was supported by European Union Digital Europe project no. 101195233 (OpenEuroLLM) and Horizon Europe project no. 101070350 (HPLT). Jindřich L. was supported by the CUNI project `PRIMUS/23/SCI/023` and project `CZ.02.01.01/00/23_020/0008518` of the Czech Ministry of Education.

Computational resources were partially provided by the e-INFRA CZ project (`ID:90254`), supported by the Ministry of Education, Youth and Sports of the Czech Republic. The work has been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure and Czech National Corpus, supported by the Ministry of Education, Youth and Sports of the Czech Republic (projects `LM2023062` and `LM2023044`).

References

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is ChatGPT that easy to detect? In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, pages 14–27, Paris, France. ATALA.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *Preprint*, arXiv:2305.07969.

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.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh

Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ben Hamner, Jaison Morgan, lynnvandev, Mark Shermis, and Tom Vander Ark. 2012. The hewlett foundation: Automated essay scoring. `https://kaggle.com/competitions/asap-aes`. Kaggle.

Adam Hájek and Aleš Horák. 2024. Czegpt-2–training new model for czech generative text processing evaluated with the summarization task. *IEEE Access*, 12:34570–34581.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Yang Jiang, Jiangang Hao, Michael Fauss, and Chen Li. 2024. Detecting chatgpt-generated essays in a large-scale writing assessment: Is there a bias against non-native english speakers? *Computers & Education*, 217:105070.

Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence*, pages 488–499, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, Maja Popović, and Mariya Shmatova. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation*, pages 1–45, Abu Dhabi. Association for Computational Linguistics.

Michal Křen, Václav Cvrček, Jan Henyš, Milena Hnátková, Tomáš Jelínek, Jan Kocek, Dominika Kováříková, Jan Křivan, Jiří Milička, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Jana Šindlerová, and Michal Škrabal. 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.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7):100779.

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

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Jiří Milička, Anna Marklová, Ondřej Drobil, and Eva Pospíšilová. 2025. Learning to detect AI texts and learning the limits. *PLOS ONE*, 20(10):e0333007. Publisher: Public Library of Science.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Ayat A. Najjar, Huthaifa I. Ashqar, Omar A. Darwish, and Eman Hammad. 2025. Detecting ai-generated text in educational content: Leveraging machine learning and explainable ai for academic integrity. *Preprint*, arXiv:2501.03203.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. *arXiv preprint arXiv:1910.00353*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024b. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor

Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Karel Šebesta, Zuzanna Bedřichová, Kateřina Šormová, Barbora Štindlová, Milan Hrdlička, Tereza Hrdličková, Jiří Hana, Alexandr Rosen, Vladimír Petkevič, Tomáš Jelínek, Svatava Škodová, Marie Poláčková, Petr Janeš, Kateřina Lundáková, Hana Skoumalová, Klement Šťastný, Šimon Sládek, and Piotr Pierscieniak. 2012. AKCES 3. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Karel Šebesta, Hana Goláňová, Jana Letafková, and Blanka Jelínková. 2016. AKCES 1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Milan Straka, Jakub Náplava, Jana Straková, and David Samuel. 2021. Robeczech: Czech roberta, a monolingual contextualized language representation model. In *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6–9, 2021, Proceedings*, page 197–209, Berlin, Heidelberg. Springer-Verlag.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

Vít Suchomel, Jan Michelfeit, and Jan Pomikálek. 2014. Text tokenisation using unitok. In *Eighth Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 71–75, Brno. Tribun EU.

Teo Susnjak and Timothy R. McIntosh. 2024. Chatgpt: The end of online exam integrity? *Education Sciences*, 14(6).

Daniel Varab and Natalie Schluter. 2021. MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv preprint arXiv:2305.18226*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

A Dataset Generation Details

Listings 1, 2, and 3 show the prompts used to generate the complement for **SYNv9**$_{\text{AUTH}}$, **WIKI**$_{\text{AUTH}}$, and **NEWS**$_{\text{AUTH}}$, respectively.

```
[CS]
Napište dalších 2000 slov tohoto textu.
Pište pouze samotný text.

<text sample>

[EN]
Write the following 2000 words of this
text. Write the actual text only.

<text sample>
```

Listing 1: Prompt for the synthetic text generation (only the `CS` version was used). The `<text sample>` is either the first paragraph, if its length is between 100 and 1000 characters, or the first n sentences ended by a full stop, such that $n \in \mathbb{N}$ is the smallest number that makes the number of characters in the sample at least 150.

```
[CS]
Napište Wikipedia článek na téma
<article name>

[EN]
Write a Wikipedia article on the topic
<article name>
```

Listing 2: Prompt for the synthetic Wikipedia articles generation (only the `CS` version was used). We substituted the <article name> for the corresponding article name from the crawled articles.

```
[CS]
Napište novinový článek na téma
'<article name>'

[EN]
Write a news article on the topic
'<article name>'
```

Listing 3: Prompt for the synthetic news articles generation (only the `CS` version was used). We substituted the <article name> for the corresponding article name from the selected news articles. Compared to the prompt for Wikipedia, the news article names were more complex, so we introduced quotes to distinguish them from the rest of the prompt.

B L1 Languages of Non-Native Speakers of Czech

The distribution of native (L1) languages of the authors of **NONNATIVE** is the following: ru: 123 (27.3%), zh: 70 (15.6%), ar: 38 (8.4%), ja: 31 (6.9%), de: 24 (5.3%), pl: 24 (5.3%), en: 23 (5.1%), fr: 21 (4.7%), ko: 19 (4.2%), bg: 13 (2.9%), it: 11 (2.4%), el: 9 (2.0%), hu: 8 (1.8%), fi: 6 (1.3%), nl: 6 (1.3%), vi: 5 (1.1%), mo: 5 (1.1%), uk: 4 (0.9%), sr: 4 (0.9%), sk: 2 (0.4%), be: 2 (0.4%), uz: 1 (0.2%), no: 1 (0.2%).

For the **NONNATIVEC1** dataset, the distribution is the following: ru: 17 (58.6%), bg: 3 (10.3%), sr: 2 (6.9%), de: 2 (6.9%), sk: 2 (6.9%), ja: 2 (6.9%), vi: 1 (3.4%).

C Token-Level Entropy Analysis

We analysed the entropy of selected texts from non-native speakers qualitatively to understand their increased entropy, and concluded that grammar errors have a prominent role in this phenomenon. We show an illustrative example in Figure 5. Notice that for most correct words split into multiple tokens, the first token is often difficult to predict, yet the remaining tokens are quite predictable from the context. This, however, does *not* hold for misspelt words, in which even the later tokens have a low probability.

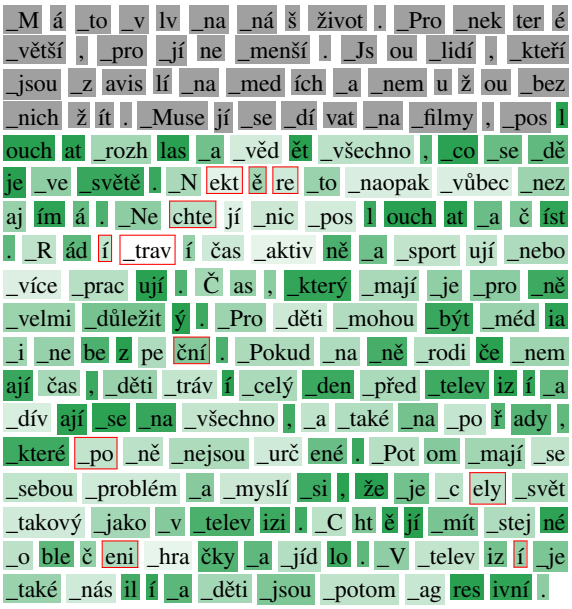

Figure 5: Tokens' contributions to entropy, written by a non-native speaker. Darker tokens have higher likelihoods. Start-word tokens begin with an underscore (_). The first 50 tokens (in grey) are used to introduce the context, and their likelihood is not measured. Non-introductory tokens with spelling or grammatical errors have red borders.

D Naïve Bayes Details

In this section, we discuss the training details of the TF-IDF Naïve Bayes detector. The detextor was trained on the **SYNV9**$^{\text{TRAIN}}$ dataset. For text vectorisation, we used the `TfidfVectorizer` from the `scikit-learn` library (Pedregosa et al., 2011). For the NB implementation, we used the `MultinomialNB` from the same library. Other than the `unitok` tokeniser, we used the default parameters. The vocabulary size (feature vector dimension) was 162 109.

We further experimented with using the RobeCzech tokeniser instead of `unitok` and got comparable results, shown in Table 6. The vocabulary size was 49 998.

E RobeCzech Details

In this section, we describe in detail the architecture, training procedure and data augmentation of the RobeCzech classifier.

E.1 Architecture

The architecture is based on the architecture for sentiment analysis described by Straka et al. (2021, Sec. 4.6). The prediction works as follows:

Dataset	Acc	FPR	FNR	Unk
SYNv9$^{\text{TRAIN}}$	98.7	0.6	2.1	0.0
SYNv9$^{\text{VAL}}$	98.8	0.3	2.0	0.1
SYNv9$^{\text{VAL}}_{\text{4OMINI}}$	99.0	–	1.0	0.0
SYNv9$^{\text{VAL}}_{\text{LLAMA}}$	72.4	–	27.6	0.1
WIKI	80.3	4.9	39.8	0.5
NEWS	96.7	6.3	0.3	0.2
NONNATIVE	93.1	6.9	–	0.3
NONNATIVEC1	65.5	34.5	–	0.2
NATYOUTH	93.6	6.4	–	0.2
NATADV	79.3	20.7	–	0.2
ABS2020	39.5	60.5	–	0.2
ABSNEW	38.5	61.5	–	0.2

Table 6: Evaluation results of the TF-IDF NB classifier with the RobeCzech tokeniser. Values are shown as per cent (%). The 'UnkR' column corresponds to the ratio of out-of-vocabulary tokens in the dataset (also in per cent).

1. The input text is tokenised, and special tokens are added; importantly, the `[CLS]` token at the beginning.

2. The tokens are passed through RobeCzech, and the output of the last hidden layer (i.e. the contextualised embeddings) is extracted.

3. The embedding of the `[CLS]` token is linearly projected to dimension 1. The rest of the embeddings are disregarded.

4. The linear projection is followed by a sigmoid activation, resulting in output $\in [0, 1]$ – the probability of positive classification.

The architecture is illustrated in Figure 6.

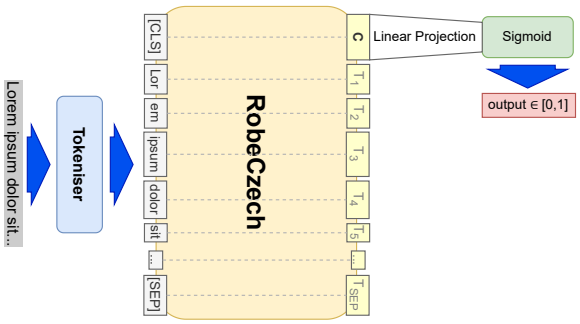

Figure 6: The architecture of the RobeCzech classifier.

E.2 Training

We trained the classifier on the **SYNv9$^{\text{TRAIN}}$** dataset with no text normalisation. We used the **SYNv9$^{\text{VAL}}$** dataset for hyperparameter tuning. We trained the model on a single NVIDIA RTX A4000 GPU (16 GB VRAM). We used the PyTorch (Paszke et al., 2019) implementation of the standard training procedures.

The training procedure consisted of three phases: in the first phase, we froze the RobeCzech parameters and only trained the linear projection layer (classification head) at a constant learning rate. In the second phase, we unfroze the RobeCzech weights and trained them with a linear learning rate warmup from 0 to a specified value. Finally, in the third phase, we kept the weights unfrozen and trained with cosine decay to 0.

We used the following hyperparameters in all the phases: batch size: 32, optimiser: AdamW (with the default $\beta_1 = 0.9$, $\beta_2 = 0.999$), weight decay: 10^{-3}, label smoothing: 0.1.

The classification head was trained by itself for one epoch at a learning rate 5×10^{-4} (10^{-3} with RDA). We were able to reach the accuracy of 89.80% on **SYNv9$^{\text{VAL}}$** after this epoch alone. Next, we trained the whole model with a linear learning rate warmup from 0 to 3×10^{-7} over one epoch. Finally, the model was trained for an additional three epochs (1 epoch with RDA) with cosine learning rate decay to 0.

E.3 Random Augmentation

In order to make the classifier more robust against rare characters, we introduced random data augmentation (RDA). The process first employs the `sacremoses`[16] punctuation normaliser and then adds random Unicode and whitespace noise.

Adding random Unicode noise involves inserting random 'words' – sequences of randomly generated printable Unicode symbols. First, the number of words to add is determined: for a sequence of w words and the expected inflation factor of 0.02, the number of added words will be $\max(0, \lfloor x \rceil)$, where $x \sim \mathcal{N}\left(0.02w, \frac{0.02w}{5}\right)$. Next, each word is generated by determining its length as $\max(0, \lfloor y \rceil)$, where $y \sim \mathcal{N}(1, 1)$, and generating such a sequence of characters. The words are then inserted into positions generated at random, with repetition.

After adding random words, we join both the original and the inserted words with random whitespace. With the probability of 0.97, we use a single space character. Otherwise, we use a sequence of $\max(1, \lfloor z \rceil)$ where $z \sim \mathcal{N}(1, 0.2)$, whitespace characters from the following list: `\n`, `\t`, `\r\n`, `\n\n`, `\r\n\r\n`, and `\u00a0` (the *non-breaking space* – ` `).

[16] https://github.com/hplt-project/sacremoses

F Entropy Distributions

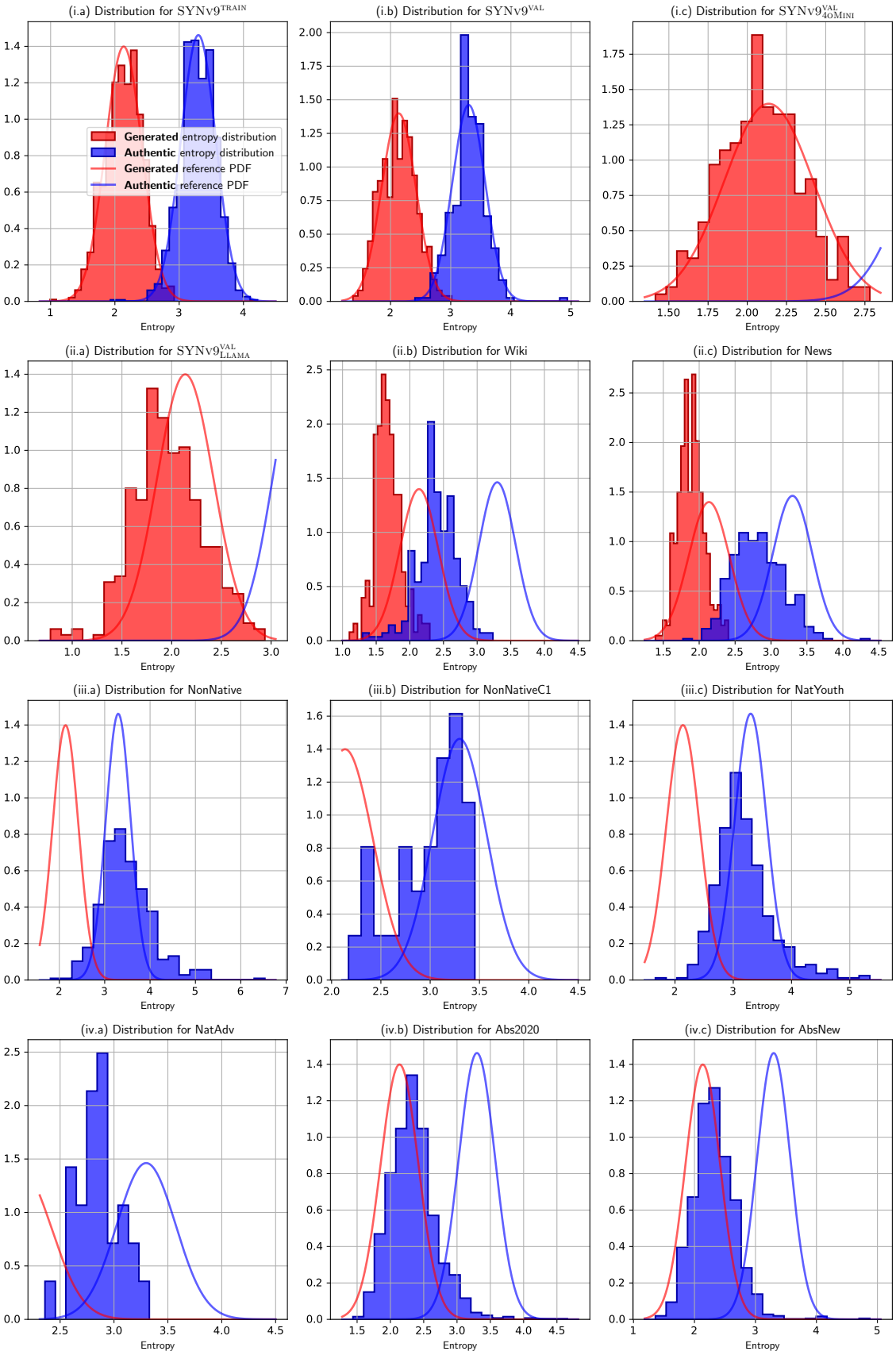

Figure 7: Entropy distribution densities of all the datasets compared to the density fitted on **SYNv9TRAIN**.

G Correlation Heatmaps

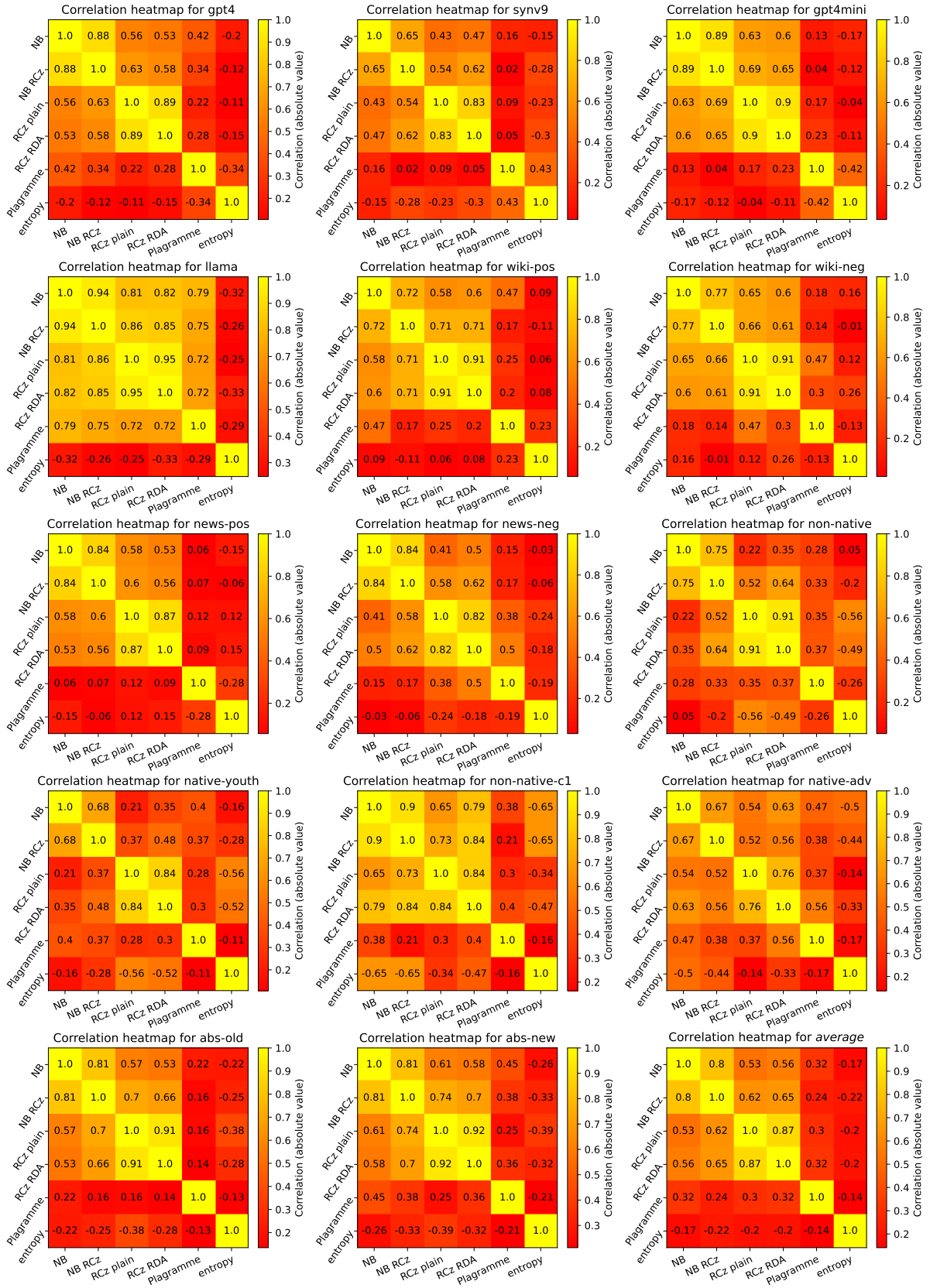

Figure 8: Correlation heatmap for the compared models, for each dataset. Key: NB: naïve Bayes detector; NB RCz: naïve Bayes detector with the RobeCzech tokeniser; RCz plain: RobeCzech detector with no augmentation, RCz RDA: RobeCzech detector with random data augmentation.