You are an LLM teaching a smaller model everything you know: Multi-task pretraining of language models with LLM-designed study plans

Wiktor Kamzela[1] **and Mateusz Lango**[1,2] **and Ondřej Dušek**[2]
[1]Poznan University of Technology, Faculty of Computing and Telecommunications, Poznan, Poland
[2]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
wiktor.kamzela@student.put.edu.pl, {lango,odusek}@ufal.mff.cuni.cz

Abstract

This paper proposes a multi-task pre-training of language models without any text corpora. The method leverages an existing Large Language Model (LLM) to generate a diverse corpus containing training data for 56 automatically designed tasks and uses generated labels to enhance the training signal. The method does not rely on hidden states or even output distributions of the teacher model, so may be employed in scenarios when the teacher LLM is available only through an API. The conducted experiments show that models trained on the proposed synthetic corpora achieve competitive or superior performance compared to those trained on same-sized human-written texts.

1 Introduction

Pretraining of language models (LMs) typically relies on massive text corpora collected from the web, books, and other sources (Gao et al., 2020; Bai et al., 2023). While this paradigm has proven highly effective for building large language models (LLMs), it also poses a significant challenge: training requires enormous computational resources to process large datasets. This limitation has sparked research interest in approaches that reduce data requirements, such as training models on smaller corpora (Hu et al., 2024) or leveraging knowledge distillation from already trained, larger models (Gu et al., 2024). Knowledge distillation, however, typically assumes access to the teacher model's hidden states, parameter values, or output distributions, which is rarely possible when the model is exposed only through an API (Xu et al., 2024).

A parallel line of research has explored the use of LLMs to generate synthetic data for model fine-tuning. Prior work has shown promising results in tasks such as text classification (Li et al., 2023), data augmentation (Long et al., 2024), and instruction tuning (Li et al., 2024). To the best of our knowledge, however, synthetic data generation has not yet been applied to *pretraining* language models. This raises two key challenges. First, LLMs tend to produce similar outputs from the same data generation prompt, making it difficult to obtain the level of diversity required for pretraining. Second, achieving strong performance on small datasets requires more efficient training techniques.

In this paper, we address these challenges by proposing multi-task pre-training of language models using an LLM-designed study plan – synthetic data that is not only automatically generated, but also composed of tasks picked by a teacher LLM. First, we instruct a teacher LLM to design a *study plan* for a smaller model, with the goal of teaching the smaller model how to solve all NLP tasks that an LLM should be able to handle. We then let the LLM iteratively generate a *dataset for each task* indicated in the previous step. This task-oriented approach to synthetic data generation, combined with the additional prompt extension strategies proposed, enhances the diversity of the output data and provides multiple synthetic labels for each text. The generated labels provide an opportunity to enrich the training signal for the language model through our proposed *multi-task loss*, which, in addition to the standard masked language modelling (MLM) objective, incorporates multiple text classification and sequence tagging losses.

The experimental evaluation performed on SuperGLUE (Wang et al., 2019) and BLiMP (Warstadt et al., 2020) benchmarks indicates that language models pretrained on synthetic data generated by the proposed technique perform competitively compared to models trained using human-written texts of the same size. Our models obtain the best average performance across both benchmarks among models trained on small corpora of 1M words. For 10M-word training corpora, our models perform best on fine-tuned downstream tasks of SuperGLUE, while models train on human-written data are better on BLiMP.

Proceedings of the First BabyLM Workshop, pages 469–487

This paper describes our submission to the interaction track of BabyLM Challenge 2025 (Charpentier et al., 2025). The model pretrained on a small 1M words multi-task corpora is publicly available at https://huggingface.co/Wector1/Multitask-pretraining_1M.

2 Problem statement

The goal of the presented method is to train a language model without relying on any preexisting text corpora. Instead, a selected large language model (LLM) is used as a teacher, but its weights, hidden states, or output distributions are not revealed to the student model. The teacher model therefore generates synthetic training data, which is then used to train the student model.

3 Task-oriented data generation

Our data generation pipeline consists of two fully automatic stages: (1) *study plan design* (selection of target NLP tasks) and (2) *generation of training examples* for each training task. In the study plan step, a teacher LLM enumerates desirable NLP tasks, designs the corresponding annotation schemas (i.e., list of target classes/tags) and constructs prompts that will generate training data following the schemas. The example generation step runs the provided prompts and diversifies them by adding requests to generate examples of a given class, a given difficulty level, or containing selected words. The overview of the data generation process is presented in Fig. 1.

3.1 Task generation

The teacher LLM is asked to design a study plan for a smaller LLM to teach the student everything it "knows". The study plan is generated in four iterations, each time asking the teacher to create a study plan for one of the four "lessons" (NLP task types): *text classification*, *text pair classification*, *sequence tagging*, and *text generation*. Apart from instructing LLM to focus on English-only tasks, the prompt requests the teacher to build a diversified list of tasks and to avoid confusion with other task categories. All prompts are provided in App. A.

Next, for each suggested task (other than text generation tasks), the LLM is asked to generate an annotation schema, containing the list of classes and their descriptions. Finally, the teacher is instructed to design a list of prompts that would make an LLM generate a dataset for a given task with

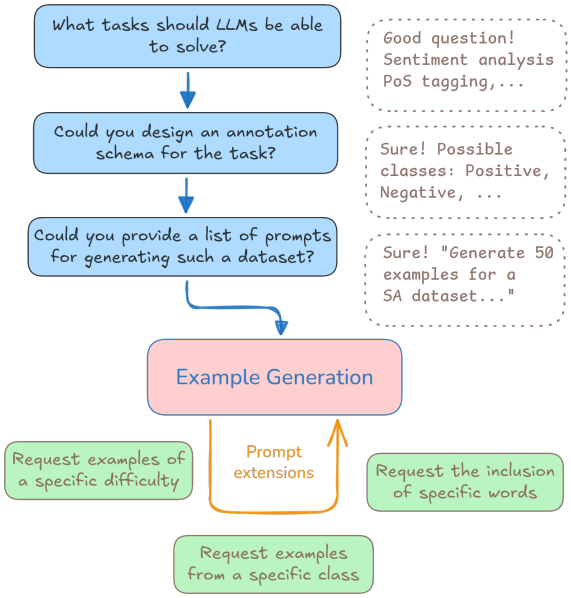

Figure 1: An overview of our data generation strategy for large language model pretraining.

the given annotation schema. The final result of the process is a list of tasks with the following attributes: name, description, task type (one of the four listed above), a list of classes/tags (with their definitions), and a list of multiple prompts that can generate a training dataset for said task.

3.2 Example generation

The training examples generation for each task is performed by collecting LLM responses for prompts generated in the previous step. As prompts are designed automatically, to avoid potential confusion during generation, we additionally used a system prompt that contains the task description, input-output specification and the instruction to respond with 50 examples.

A major issue when generating a large dataset with LLM is obtaining diverse examples. To this end, we designed three prompt extenders: *difficulty extender*, *label extender* and *vocabulary extender*. Each extender worked by appending a sentence with additional instructions to the original prompt.

The *difficulty extender* asks for easy, medium and hard to classify examples. The *label extender* specifically requests examples belonging to a single selected class. The *vocabulary extender* is the most advanced: it tracks the vocabulary in already generated examples and requests texts containing at least one of five target words. These words are selected as the least frequent in the already generated samples, except for words occurring less

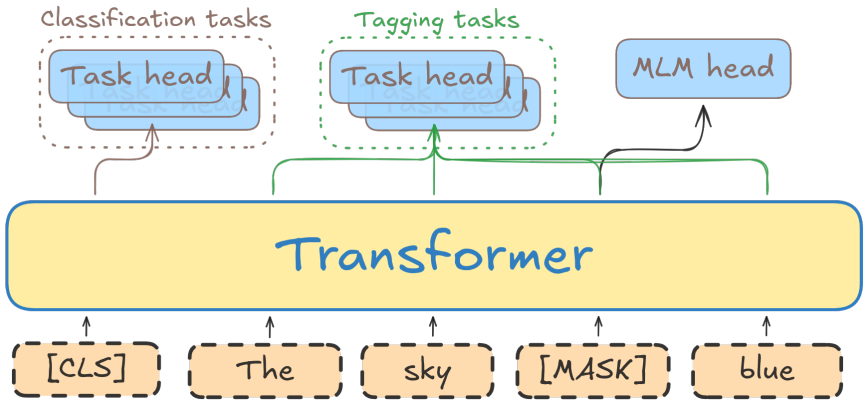

Figure 2: An overview of the proposed architecture of task-augmented pretraining of a language model.

than three times (to avoid noise). This allows for a gradual expansion into more complex vocabulary.

3.3 Dataset postprocessing and multi-task labeling

The data generated from the previous steps is a collection of datasets for different tasks, where each component dataset contains only one type of label. To fully embrace the potential of multi-task pretraining, we generate labels/tags *for all tasks* for each given input. For instance, a given sentence originally generated for the task of spam classification will additionally obtain tags for part-of-speech classification, sentiment analysis, etc. To this end, we asked the teacher LLM to act as a classifier/-tagger for each given task (except text generation tasks) and to provide labels for the whole dataset. See prompt in App. A.

Finally, the generated dataset undergoes a simple filtering, consisting of deduplication and removing all instances that contain fewer than 3 words or contain non-English characters. Additionally, any inconsistent synthetic annotation (e.g., sequence tag lists of incorrect lengths or labels outside of the designed schema) is discarded, i.e., each instance in the final data includes labels for most but not necessarily all NLP tasks.

4 Task-augmented pretraining

To take advantage of the generated preprocessing data that contains labels for artificially constructed tasks, we propose a task-augmented pretraining method that modifies the standard transformer architecture by adding multiple classification/tagging heads. Each task head is associated with a task loss function, which enriches the standard MLM loss with an additional training signal, allowing for training with smaller datasets. The overview of the architecture is presented in Fig. 2.

Language modeling The architecture of our model is a bidirectional transformer with the input format following that of BERT. The input sentence begins with a start token `[CLS]` and finishes with `[SEP]`. If the input was generated from a text pair classification task, the input texts are also separated with `[SEP]` token. Note that in text generation tasks, the sentences are not separated by any special tokens as they are only used for standard MLM objective.

During pretraining, the input to the model is perturbed using default Masked Language Modeling parameters in HuggingFace. More concretely, 15% of input tokens are masked: 80% of them are replaced by a special `[MASK]` token, 10% is replaced with a random token and the remaining 10% is left unchanged. An MLM classification head is attached to the output embedding of each masked word that predicts the word at the given position.

$$\mathcal{L}_{MLM} = -\sum_{x_i \in Masked} \log P(x_i | Masked(x)_i)$$

Task heads For each task (except text generation tasks) present in the dataset, a new classification head is constructed, which takes as input the output of the final layer of the transformer network. For efficiency, the tasks are performed on the same, i.e. masked, input as MLM.

For tagging tasks, the corresponding task head is applied to the representation of every input token.

$$\mathcal{L}_{tag\_task_j} = -\sum_{x_i} \log P(y_i^{(j)} | Masked(x)_i)$$

For text and text pair classification tasks, the classification head is applied to the `[CLS]` token.

$$\mathcal{L}_{class\_task_j} = -\log P(y_i^{(j)}|Masked(x)_{[CLS]})$$

Note that every input has multiple labels corresponding to different tasks, and all classification heads are applied simultaneously. However, in the case of the data generation process failing to generate labels for some tasks, the task heads are dynamically detached from the transformer (i.e., only task heads for which labels are available are used).

Multi-task loss The final loss optimized by the model during pretraining is a weighted sum of masked language modeling loss and task losses.

$$\begin{aligned}\mathcal{L} = \; & \mathcal{L}_{MLM} \\ & + w_J J^{-1} \sum_j \mathcal{L}_{class\_task_j} \\ & + w_K K^{-1} \sum_k \mathcal{L}_{tag\_task_k}\end{aligned}$$

where J is the number of text and text pair classification tasks and K is the number of sequence tagging tasks, w_J and w_K are hyperparameters of the loss function.

5 Experiments

5.1 Experimental setup

Data generation The data generation pipeline was executed with Llama 3.1 8B Instruct (Grattafiori et al., 2024) as the teacher LLM. The pipeline was implemented with vLLM library (Kwon et al., 2023), during the generation repetition penalty was set to 1.1, top_k and top_p parameters to 40 and 0.9, respectively.

To fully automate the data generation process, after generating artifacts such as the list of tasks in free-text, the LLM was asked to reformat its response into JSON, with a structured output format enforced as provided by VLLM library. We find that this two-step generation resulted in more diverse samples than directly enforcing the generation of structured output. This generation strategy was also used for all the described interactions with LLM.

Three versions of the dataset were generated:

- *Multi-task corpus* – corpus containing 56 diverse tasks proposed by the teacher model. The full list of tasks is given in App. B. Some of the tasks designed by the teacher model would most probably not be proposed by a human expert, e.g. a text summarization task which belongs to the text classification category (detect if a given text is a summary) or caption writing for images (our model is text-only). Nevertheless, we decided to keep them, assuming that LLM will be consistent while producing and annotating such datasets, which can provide some additional signal for the student model.

- *Text generation corpus* – corpus constructed by the proposed method, but with tasks limited to the text generation category. To obtain a longer list of tasks, the model was prompted to provide an exhaustive list of topics that should be contained in the training corpora of an LLM. This resulted in 58 text generation tasks (talk and discuss a given topic). The rest of the pipeline (i.e. prompt construction, example generation) was performed as previously described.

- *Vocabulary-controlled corpus* – corpus generated identically as "Text generation corpus", but half of it was generated by the LLM using only 5k different tokens (all other tokens were masked from the prediction head). The idea was that providing a lot of training data on a limited vocabulary would help the model better learn the grammar and the representation of the most frequent words. The tokens were selected by running the tokenizer on the whole English Wikipedia corpus[1].

The datasets were generated in two sizes: 1M and 10M words. The datasets are constructed by selecting 1000 examples (10k for the 10M version) from every tagging/text classification task, and the rest consists of texts from the text generation tasks.

Model architecture Our model architecture is based on ModernBERT (Warner et al., 2025) implementation from HuggingFace library (Wolf et al., 2020). Two model sizes were tested:

- 149M[2] architecture of original ModernBERT-small with default parameters, except for the context size set to 256.

[1] https://github.com/GermanT5/wikipedia2corpus

[2] The model size does not include the size of task heads, as they are discarded after pretraining.

	D. size	Dataset	Epochs	BoolQ	MNLI	MultiRC	RTE	WSC	MRPC	QQP	BLiMP	S. GLUE	Average
ModernBERT 149M	1M	Text gen.	10	0.686	0.452	0.664	0.518	0.635	0.701	0.690	55.22	0.621	58.65
		Multi-task	10	0.713	0.444	0.669	0.554	0.615	0.730	0.715	56.80	0.634	60.11
		Vocab. c.	10	0.689	0.451	0.665	0.532	0.692	0.706	0.692	53.12	0.633	58.19
		Text gen.	100	0.689	0.459	0.666	0.554	0.654	0.706	0.713	55.56	**0.634**	59.50
		Multi-task	100	0.696	0.429	0.667	0.583	0.615	0.730	0.700	**57.82**	0.631	**60.48**
		Vocab. c.	100	0.691	0.448	0.665	0.547	0.635	0.721	0.704	56.09	0.630	59.55
		Text gen.	500	0.698	0.452	0.664	0.540	0.635	0.676	0.731	55.82	0.628	59.31
		Multi-task	500	0.692	0.416	0.660	0.532	0.615	0.716	0.705	57.08	0.619	59.51
		Vocab. c.	500	0.703	0.434	0.675	0.540	0.615	0.750	0.705	55.59	0.632	59.39
	10M	Text gen.	10	0.708	0.494	0.665	0.525	0.654	0.745	0.744	61.84	0.648	63.32
		Multi-task	10	0.701	0.453	0.666	0.576	0.635	0.730	0.729	64.38	0.641	64.25
		Vocab. c.	10	0.704	0.485	0.674	0.568	0.635	0.745	0.740	61.78	0.650	63.39
		Text gen.	50	0.698	0.526	0.670	0.561	0.654	0.730	0.767	63.06	0.658	64.44
		Multi-task	50	0.707	0.445	0.673	0.547	0.615	0.696	0.733	**65.19**	0.631	64.14
		Vocab. c.	50	0.702	0.509	0.673	0.619	0.654	0.735	0.758	63.76	**0.664**	**65.10**
ModernBERT 39M	1M	Text gen.	10	0.680	0.429	0.652	0.525	0.654	0.686	0.709	54.12	0.619	58.03
		Multi-task	10	0.691	0.434	0.653	0.540	0.635	0.755	0.715	56.15	0.632	59.66
		Vocab. c.	10	0.677	0.445	0.665	0.525	0.635	0.706	0.718	52.48	0.624	57.46
		Text gen.	100	0.684	0.458	0.666	0.561	0.654	0.706	0.685	55.87	0.631	59.47
		Multi-task	100	0.687	0.437	0.658	0.532	0.654	0.725	0.710	57.63	0.629	**60.28**
		Vocab. c.	100	0.683	0.440	0.669	0.518	0.673	0.701	0.708	56.42	0.627	59.58
		Text gen.	500	0.683	0.433	0.662	0.532	0.692	0.730	0.720	55.75	**0.636**	59.68
		Multi-task	500	0.680	0.414	0.649	0.583	0.615	0.676	0.717	**58.60**	0.619	60.25
		Vocab. c.	500	0.696	0.438	0.666	0.561	0.635	0.706	0.712	55.25	0.631	59.15
	10M	Text gen.	10	0.686	0.459	0.658	0.525	0.654	0.706	0.727	60.07	0.631	61.57
		Multi-task	10	0.683	0.445	0.665	0.525	0.635	0.721	0.727	63.30	0.629	63.08
		Vocab. c.	10	0.678	0.464	0.666	0.576	0.654	0.701	0.725	59.31	0.638	61.54
		Text gen.	50	0.684	0.514	0.676	0.590	0.654	0.721	0.761	61.45	**0.657**	63.58
		Multi-task	50	0.693	0.454	0.667	0.504	0.596	0.711	0.728	**65.08**	0.622	**63.63**
		Vocab. c.	50	0.686	0.484	0.669	0.561	0.635	0.706	0.747	62.49	0.641	63.30

Table 1: Results of evaluation of trained models on BLiMP and SuperGLUE (S. GLUE) benchmark. The best results for a given model and data size are bolded.

- 39M ModernBERT architecture with halved hidden size to 384, intermediate size to 576, and 16 layers.

Training details Models were trained with AdamW optimizer with 128 batch size. Learning rate followed the cosine schedule with 500 warmup steps and a learning rate of 0.0003. The weights of the multi-task loss (see Sec. 4) were selected to $w_J = w_K = 0.5$. A small weight decay of 0.01 was applied.

The number of epochs depended on the size of the dataset. The smaller 1M dataset was tested with 10 epochs (10M tokens seen during training), 100 epochs (100M tokens) and 500 epochs (500M tokens). The larger 10M dataset was tested with 10 epochs (100M tokens) and 50 epochs (500M tokens).

Evaluation Our evaluation follows the evaluation framework provided by the BabyLM Challenge organizers (Charpentier et al., 2025). More concretely, we evaluated our model's language understanding capabilities using the SuperGLUE benchmark, encompassing the tasks BoolQ, MNLI, MultiRC, RTE, WSC, MRPC, and QQP (Wang et al., 2019). For each task, pretrained models were fine-tuned with default parameters provided by BabyLM organizers without hyperparameter tuning. Additionally, we benchmarked grammatical knowledge using BLiMP (the Benchmark of Linguistic Minimal Pairs), which comprises 67 subdatasets of minimal sentence pairs probing syntax, morphology, and semantics (Warstadt et al., 2020). For the convenience of model comparisons, we also report the average of BLiMP and SuperGLUE scores, with the latter multiplied by 100 for scale adjustment.

Dataset	Epochs	BoolQ	MNLI	MultiRC	RTE	WSC	MRPC	QQP	BLiMP	S. GLUE	Average
Human 1M	10	0.691	0.442	0.660	0.597	0.635	0.721	0.699	54.38	**0.635**	58.94
	50	0.681	0.425	0.658	0.554	0.673	0.706	0.696	57.66	0.628	60.21
	500	0.698	0.419	0.663	0.547	0.615	0.711	0.688	57.12	0.620	59.56
Best synthetic 1M	50	0.696	0.429	0.667	0.583	0.615	0.730	0.700	**57.82**	0.631	**60.48**
Human 10M	10	0.698	0.449	0.670	0.554	0.654	0.770	0.725	69.38	0.646	66.97
	50	0.694	0.458	0.668	0.576	0.654	0.730	0.745	**71.68**	0.646	**68.15**
Best synthetic 10M	50	0.702	0.509	0.673	0.619	0.654	0.735	0.758	63.76	**0.664**	65.10

Table 2: Results of evaluation of ModernBERT 149M trained on human text corpora (BabyLM) compared to the best model (acc. to average) trained on synthetic data of the same size.

5.2 Results

The evaluation results for models trained on data synthesized by our method are presented in Table 1.

Analyzing model performance as the average across both benchmarks, multi-task pretraining achieved the best results for all model and dataset sizes, as well as for all training durations measured in epochs. The only exception was the 149M-parameter model trained on the 10M corpus with a computation budget of 50 epochs, where training on the vocabulary-constrained corpus yielded the best average score, although it was still outperformed by multi-task pretraining on BLiMP.

The comparison between text generation and vocabulary-constrained corpora does not reveal a clear winner, as both approaches produced very similar results across all tested configurations. Likewise, we did not observe substantial performance differences between the two studied model sizes. However, the comparison of the best average results indicates that slightly better outcomes were achieved with the larger model for both corpus sizes.

Comparing models trained on the 1M corpus for 500 epochs and the 10M corpus for 50 epochs (both exposed to the same total number of tokens), we observe clear benefits from training on more diverse texts rather than repeatedly reusing the same content. BLiMP improves by 7 percentage points, while SuperGLUE increases by about 3 percentage points. Interestingly, the best average results among models trained on 1M-word corpora were obtained with a 100-epoch budget, suggesting no clear benefits from increased training time.

Comparison with pretraining on human-written corpora To compare the effectiveness of training on our synthetic datasets with training on human-written texts, we trained the 149M version of our model on BabyLM corpora (Charpentier et al., 2025) using the same model hyperparameters. We took the 10M-word version of the BabyLM corpus (denoted as Human 10M) and additionally used the first 1M tokens of it as the smaller, 1M-word version (Human 1M). Table 2 reports the evaluation results of models trained on human-written corpora, along with the best-performing synthetic-data models for each corpus size, provided as a reference.

For a corpus of one million words, models trained using Human-1M performed worse than the best model trained on synthetic data, measured as an average of GLUE and BLiMP benchmarks. On larger corpora, the models trained on human data obtained higher BLiMP score, but the model trained on synthetic data was still better for fine-tuning on downstream tasks of GLUE benchmark.

Analysis of generated data Basic statistics and vizualizations of generated datasets are presented in App. C. The multi-task dataset contains a significantly larger number of samples with shorter texts in comparison to other corpora. This is because many classification tasks operate on single sentences rather than the paragraphs or documents typically found in text corpora.

As expected, the vocabulary-controlled dataset has the smallest vocabulary, whose frequency distribution has a significantly shorter tail than those of the other studied datasets. The multi-task and text generation datasets both have fewer occurrences of high-frequency words than the human corpus and higher vocabulary diversity.

The vast majority of tasks have imbalanced label distributions. A few tasks have very long-tail distributions of label frequencies and classes with nearly zero instances. About a quarter of the examples in 1M corpora and only 4.7% of them in 10M one have labels for any sequence prediction/tagging task. This is due to the teacher LLM's failure

Dataset	BLIMP	S. GLUE	Average
Multi-task	56.80	0.634	60.11
only text classification	57.08	0.645	60.79
only pair text class.	56.85	0.639	60.38
only tagging	57.67	0.627	60.20

Table 3: Results of evaluation ModernBERT 149M trained for 10 epochs for different versions of 1M words Multi-task corpus.

to generate a sequence of labels of the expected length[3], hindering the possible gains from these tasks.

Ablation study We performed an ablation study to verify which tasks categories contribute the most to the final results. We performed experiments on 1M Multi-task corpus keeping only labels from a selected task category and training for 10 epochs the larger version of our model.

The results are presented in Table 3 and are slightly higher than those for the basic version of the multi-task corpus containing all the labels. Training only on text classification labels yields the highest improvement. This may be related to the fact that this group of tasks has the highest number of generated labels in our corpus, providing labels for almost all instances and thus making the corresponding task heads well-trained.

6 Summary

This paper introduces a method for pretraining language models entirely on synthetic data generated by a large language model (LLM) using fully automatic pipeline. The teacher model automatically design and generate datasets for diverse NLP tasks, spanning across text classification, tagging, and text generation. The additional training information comming from synthetic labels is exploited during training via the proposed multi-task loss.

Experiments with transformer-based language models on SuperGLUE and BLiMP benchmarks demonstrated that fully synthetic, automatically generated multi-task corpora can serve as an effective substitute for human text in pretraining.

Acknowledgments

This work was supported by the European Research Council (Grant agreement No. 101039303, NG-NLG) and used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

[3]The structure decoding algorithm allowed for format specification and limited the generation to lists of valid label names, but it could not control the length of the label list

Limitations

Some concerns related to training LLMs on existing text corpora are related to potential copyright and privacy issues associated with using web-scraped content and learning potential biases expressed in the data. Although the presented method do not use any existing text corpora, it exploits an LLM that was trained on web-scraped data, so the generated synthetic data may have similar issues.

This paper was limited in testing different configurations of trained models and it is highly probable that the training parameters used were not optimal.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

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.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *Preprint*, arXiv:2306.08543.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining

on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2024. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *Preprint*, arXiv:2402.13064.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *Preprint*, arXiv:2402.13116.

A Data generation prompts

The prompts used in the data generation pipeline are presented in Listings 1, 2, 3, 4, and 5.

B List of tasks in multi-task corpora

The list of tasks designed by LLM for the multi-task corpora is provided in Tab. 8. The histograms of labels for text classification, pair text classification and tagging tasks are presented in Fig. 3, 4 and 5, respectively. If the number of different labels for a task was higher than 10, the smallest classes was aggregated to "Other" class to keep the figures readable.

C Additional dataset characteristics

Basic corpora statistics for 1M datasets are presented in Table 4 and in Table 6 for 10M. Additionally, basic label statistics for 1M Multi-task dataset are provided in Table 5 and in Table 7 for 10M.

The histograms of text lengths in studied 1M corpora are presented in Fig. 6. Word frequency distributions in the studied 1M corpora are shown in Fig. 7.

```
    You are a large language model teaching a smaller language model everything you
    know. The smaller model should only cover English, so exclude anything related to
    other languages (translation, coding, language identification, etc.).

Your student's study plan contains several stages that will enable them to learn how
to perform all the tasks that you can perform.

You are currently planning a learning stage involving text classification tasks. These
tasks require a single text as input and provide a single class as output. Note that
this learning stage should not include any other tasks, such as the classification of
pairs of texts or sequences.

Please generate an exhaustive list of text classification tasks, which should provide
sufficient material for creating a versatile language model.
```

Listing 1: Task generation prompt for creating a list of text classification tasks.

```
    Generate an annotation schema of a dataset for {name} task.
Task description: {description}

Since this is a {task_type} task, an annotation scheme is simply a list of all
possible classes. The classification should be fine-grained, but the classes should be
precisely defined so there is no ambiguity in the labeling. It is better to have fewer
classes that are well defined than many classes that are not clearly defined.
```

Listing 2: Prompt for designing annotation schema for different tasks.

```
I want to generate artificial dataset for a machine learning task using an LLM. Here
are the details of the task:

===
TASK INFORMATION
The task is called: {task_name}.
Task description: {task_description}.
===

Could you give me a long list of possible prompts that would make an LLM generate
about 50 test examples (i.e. possible inputs)? The prompts should be clear and
diversified.  You can assume that the LLM is already aware of the information given in
'TASK INFORMATION' section provided above. While you can include a few examples in
some prompts, remember that your task is to create the prompt for the LLM, NOT to
generate the data.
```

Listing 3: Prompt for designing a list of prompts for a given task.

```
You are a data generation assistant. Your role is to create high-quality synthetic
data tailored to the user's specifications. You generate data that is realistic,
diverse, and suitable for tasks such as machine learning training, testing, and
simulation.

Your ONLY functionality is to generate diversified data for the following task:

Task name: {task_name}
Task description: {task_description}
Task type: {task_category}
Tag set: {task_class_list}

For each given prompt, you should respond with a list of 50 examples. {type_dsc}

Do not include explanations unless explicitly asked. Only return raw data or
structured output as specified. While generating examples, take into account the given
user prompts, but remember that producing data that follow the task specification
given above is crucial.
```

Listing 4: System prompt for the generation of training examples.

```
You are a text classifier, for the following task:

Task name: {task_definition['name']}
Task description: {task_definition['description']}
Task type: {task_definition['task_type']}
Tag set:
    {class_dsc}

For each given pair of input sentences provided by the user, classify it into one of
the following categories: {self.list_of_labels}.

For a given input, respond with a JSON object that matches the following schema:
{format_dsc} where label is one of the labels from the tag set.

Don't respond with any explanations, just return the JSON object.
```

Listing 5: Prompt used to construct a classifier for a given task.

Dataset	Samples	Total Chars	Avg Char	Median Char	Max Char	Avg Token	Median Token	Max Token
Text gen.	2529	7049642	2788	2894	9193	570	584	2510
Multi-task	38055	6576964	173	90	6096	36	18	1262
Vocab. c.	1187	5677209	4783	4698	10913	1078	1042	2094
Human	5007	5207559	1040	1046	1,531	254	254	255

Table 4: Basic characteristics of used 1M corpora. Number of samples and text length statistics measured in tokens and characters. Human corpora was provided as a free-text, without division into samples – we treated a training batch as a sample to compute these statistics.

Task	Unique Tasks	Samples With Task	Unique Labels	Total Labels
Text Classification	19	22889	263	313139
Text Pair Classification	12	12012	105	97152
Sequence Prediction	6	9891	78	81650

Table 5: Basic characteristics of task labels generated in 1M Multi-task dataset.

Dataset	Samples	Total Chars	Avg Char	Median Char	Max Char	Avg Token	Median Token	Max Token
Text gen.	25,246	70,492,774	2792	2910	9,193	570	584	2,510
Multi-task	641,324	131,874,012	206	95	7,305	43	19	1,860
Vocab. c.	11,927	56,711,626	4755	4672	12,289	1077	1043	2,338
Human	67,740	54,202,906	800	814	1,563	254	254	255

Table 6: Basic characteristics of used 10M corpora. Number of samples and text length statistics measured in tokens and characters. Human corpora was provided as a free-text, without division into samples – we treated a training batch as a sample to compute these statistics.

Task	Unique Tasks	Samples With Task	Unique Labels	Total Labels
Text Classification	19	185204	263	475454
Text Pair Classification	12	100552	105	185692
Sequence Prediction	6	30394	78	265910

Table 7: Basic characteristics of task labels generated in 10M Multi-task dataset.

Table 8: List of tasks in the generated dataset

Task Type	Task Name	#Classes	Description
text classification	Movie Review Sentiment Analysis	3	Classify movie reviews as positive or negative.
text classification	Product Review Sentiment Analysis	12	Classify product reviews as positive, negative, or neutral.
text classification	Political Speech Sentiment Analysis	3	Classify political speeches as positive, negative, or neutral.
text classification	Email Spam Classification	3	Classify emails as spam or non-spam based on their content.
text classification	Text Message Spam Classification	2	Classify text messages as spam or non-spam based on their content.
text classification	Topic Modeling	25	Classify articles into topics like science, technology, politics, sports, entertainment, etc.
text classification	Product Category Classification	58	Classify products into categories like electronics, clothing, home goods, etc.
text classification	Emotion Classification	5	Classify text as happy, sad, angry, surprised, or fearful.
text classification	Intent Classification	15	Classify text as booking a hotel room, making a reservation, asking for directions, etc.
text classification	Aspect-Sentiment Analysis	10	Identify aspects of a product (e.g., quality, price, design) and classify the sentiment towards each aspect.
text classification	Hate Speech Classification	4	Classify text as hate speech or not.
text classification	Toxic Content Classification	13	Classify text as toxic or not.
text classification	Product Recommendation	5	Classify text as recommending a product or service.
text classification	Question Type Classification	9	Classify questions as fact-based, opinion-based, or open-ended.
text classification	Text Summarization Classification	2	Classify text as a summary or not.
text classification	Fake News Classification	5	Classify news articles as fake or real.
text classification	Medical Condition Classification	17	Classify text as describing a specific medical condition.
text classification	Occupation Classification	12	Classify text as describing a particular occupation.
text classification	Location Classification	60	Classify text as describing a specific location.
text pair classification	Entailment Tasks	4	Determine if one text implies or supports another.
text pair classification	Recognizing Textual Entailment (RTE)	3	Similar to textual entailment but more challenging.
text pair classification	Question Pair Classification	10	Classify question types and answer types.
text pair classification	Text Similarity and Dissimilarity	6	Measure how similar two texts are in terms of meaning.
text pair classification	Contrasting Texts	5	Identify pairs of contrasting statements.
text pair classification	Emotion and Sentiment Analysis	10	Classify emotional tone and determine sentiment polarity.
text pair classification	Coherence and Consistency	5	Evaluate coherence and detect inconsistencies.
text pair classification	Argumentation and Debate	7	Assess argument strength and detect persuasion.
text pair classification	Factuality and Veracity	5	Verify facts and evaluate trustworthiness.
text pair classification	Identity and Intent	16	Identify authors and speakers, and infer intent.
text pair classification	Relationship and Entity Classification	27	Extract relationships and resolve coreferences.
text pair classification	Text Generation and Editing	7	Evaluate grammaticality and fluency.
sequence prediction	Part-of-Speech (POS) Tagging	17	Identify the grammatical category of each word in a sentence.
sequence prediction	Named Entity Recognition (NER)	9	Identify named entities in a sentence.
sequence prediction	Chunking or Phrase Chunking	13	Identify phrases or chunks within a sentence.
sequence prediction	Dependency Parsing	20	Analyze the grammatical structure of a sentence.
sequence prediction	Semantic Role Labeling (SRL)	12	Identify the roles played by entities in a sentence.
sequence prediction	Coreference Resolution	7	Identify pronouns and their antecedents.
text generation	Text Summarization	0	Given a long piece of text, generate a concise summary while preserving essential information.
text generation	Article Generation	0	Write an original article on a given topic, including introductory paragraphs, main content, and conclusion.
text generation	Storytelling	0	Create a short story based on a prompt, including characters, setting, plot, and resolution.
text generation	Dialogue Generation	0	Generate conversations between two or more people on a specific topic or scenario.

Task Type	Task Name	#Classes	Description
text generation	Product Description Writing	0	Craft compelling product descriptions based on product specifications, features, and benefits.
text generation	Social Media Post Generation	0	Write engaging social media posts, including captions and hashtags, for a variety of topics and platforms.
text generation	Email Response Generation	0	Respond to emails with a personalized message, addressing the sender's concerns or questions.
text generation	Chatbot Conversations	0	Engage in natural-sounding conversations with users, providing relevant information and support.
text generation	Poetry Generation	0	Create original poems based on prompts, using various forms and styles.
text generation	News Article Rewriting	0	Rewrite news articles in different tones, styles, or formats while maintaining the same facts.
text generation	Speechwriting	0	Write speeches for various occasions, such as weddings, graduations, or business presentations.
text generation	Book Reviews	0	Generate reviews of books, including summaries, analysis, and opinions.
text generation	Recipe Writing	0	Create recipes with step-by-step instructions, ingredient lists, and nutritional information.
text generation	Travel Itinerary Planning	0	Plan travel itineraries, including suggested routes, activities, and accommodations.
text generation	Mad Libs	0	Fill in missing words in a story or sentence with the correct parts of speech (e.g., noun, verb, adjective).
text generation	Creative Writing Prompts	0	Complete writing prompts that encourage creative thinking and storytelling.
text generation	Transcription	0	Transcribe spoken text into written form, maintaining accuracy and clarity.
text generation	Caption Writing	0	Write captions for images, videos, or memes, conveying the essence of the content.
text generation	Conversation Flow	0	Generate conversation flows for various scenarios, ensuring a logical and coherent discussion.
text generation	Scriptwriting	0	Write scripts for movies, plays, or TV shows, including dialogue and scene descriptions.

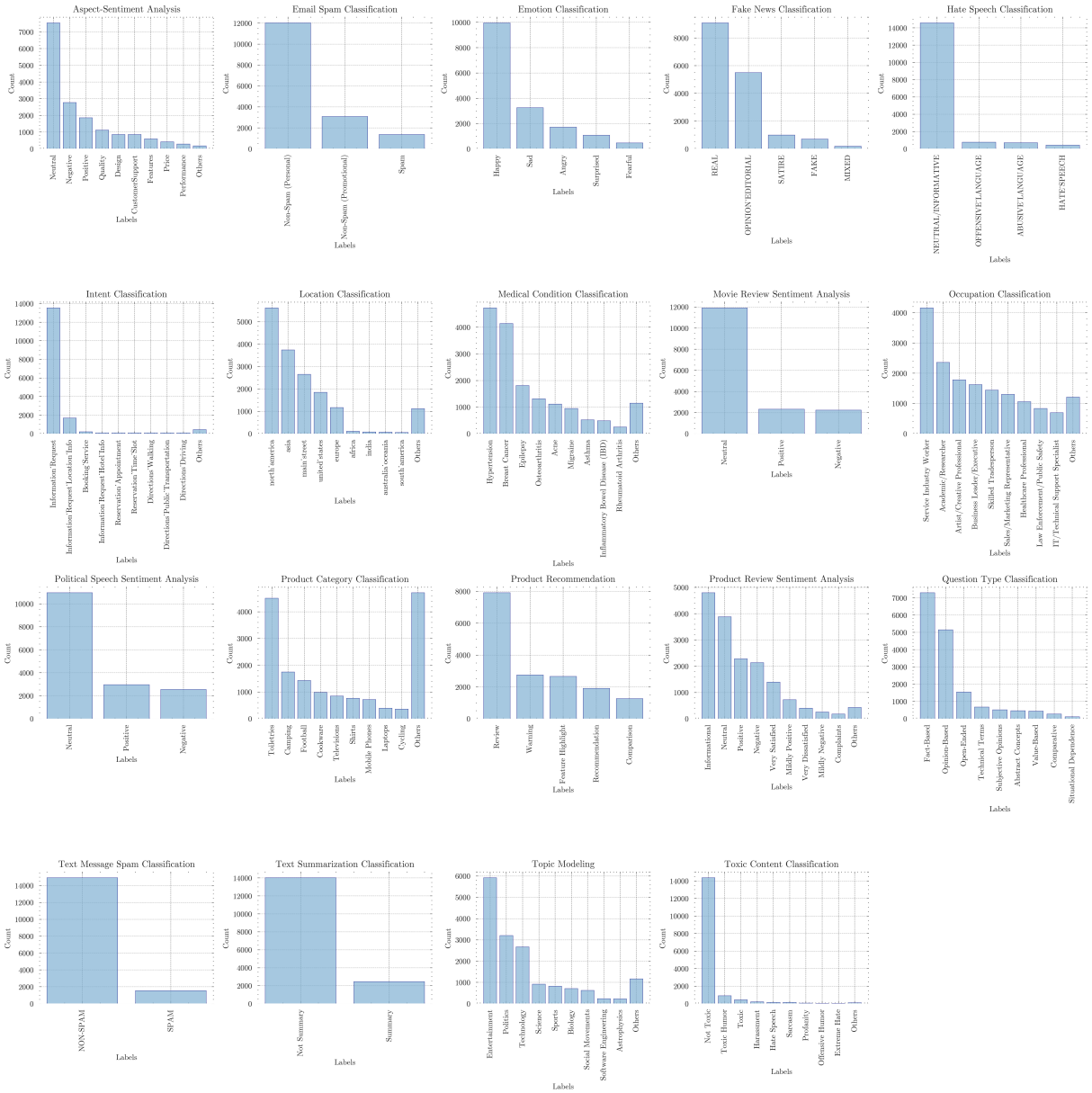

Figure 3: Histograms of labels for text classification tasks in the Multi-task corpus (1M words).

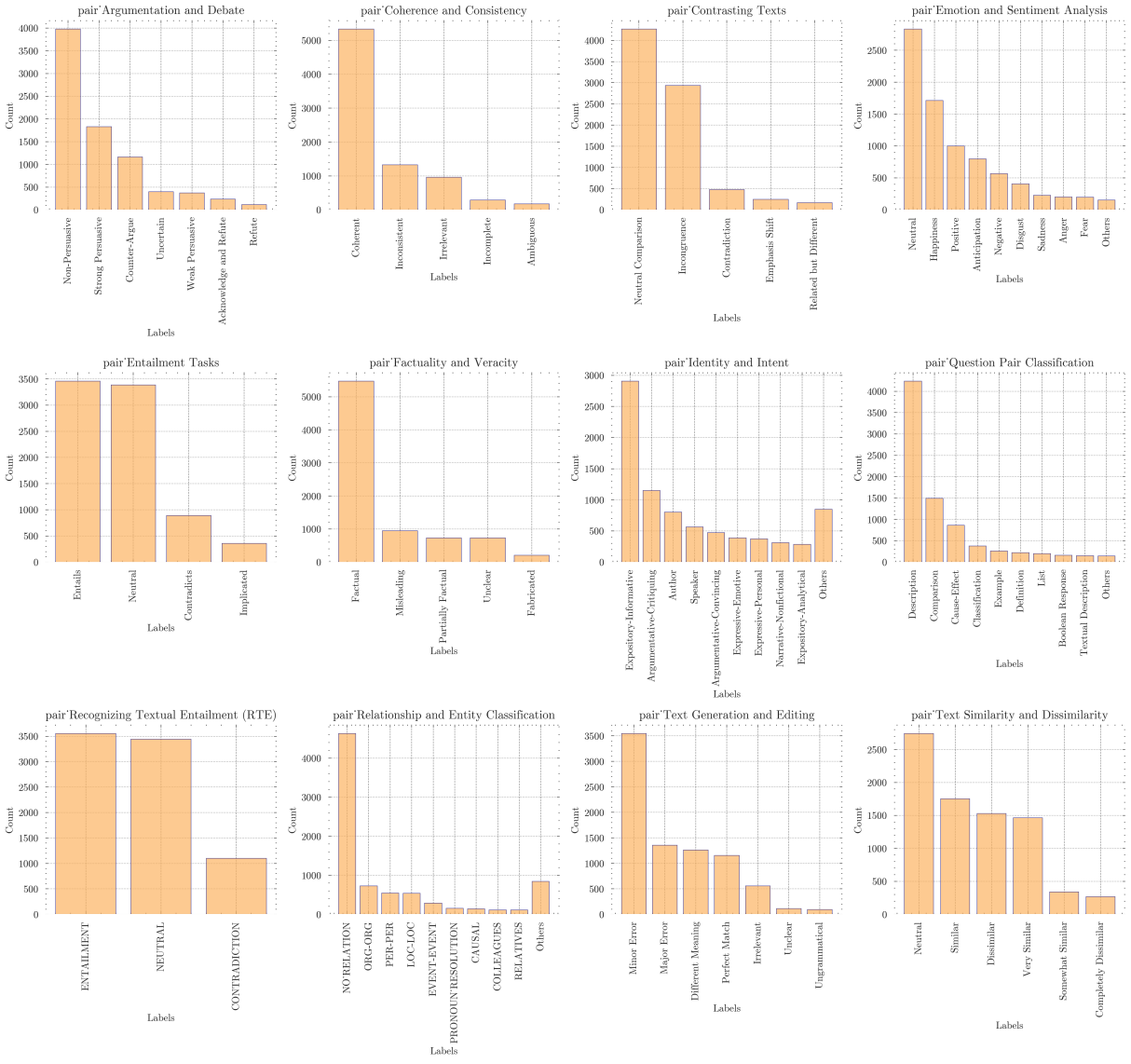

Figure 4: Histograms of labels for pair text classification tasks in the Multi-task corpus (1M words).

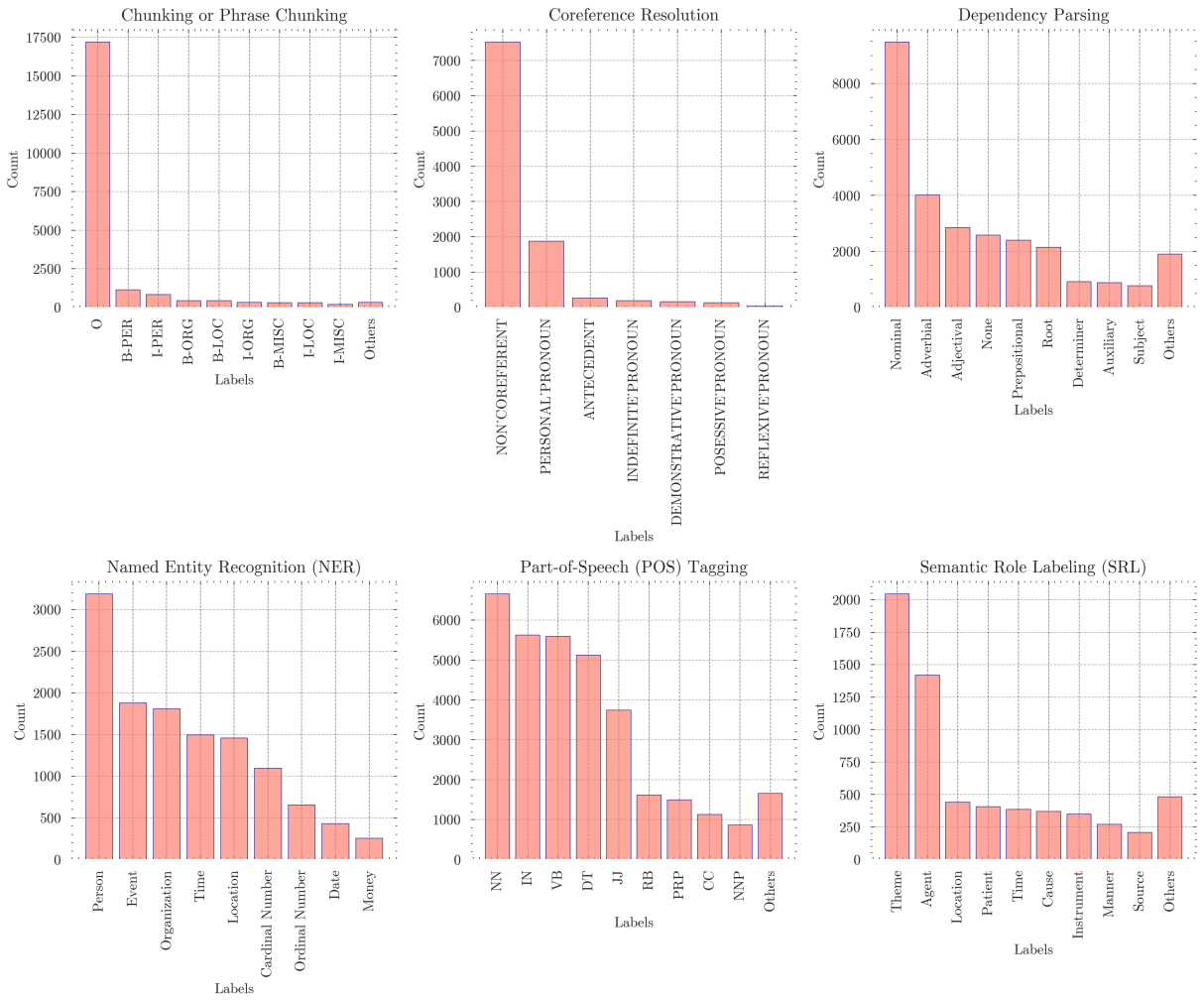

Figure 5: Histograms of labels for tagging tasks in the Multi-task corpus (1M words).

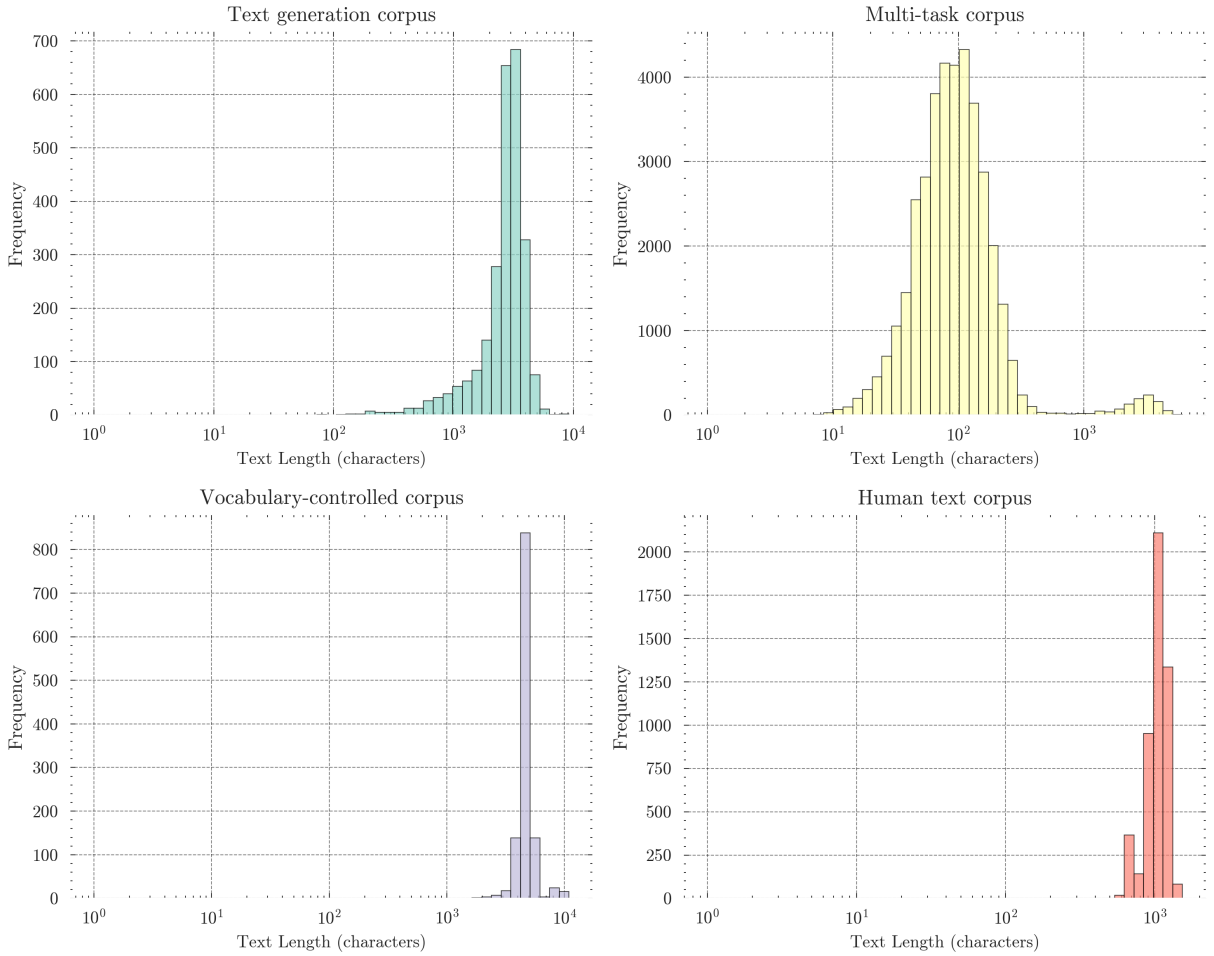

Figure 6: Histograms of text lengths (measured in characters) for different datasets.

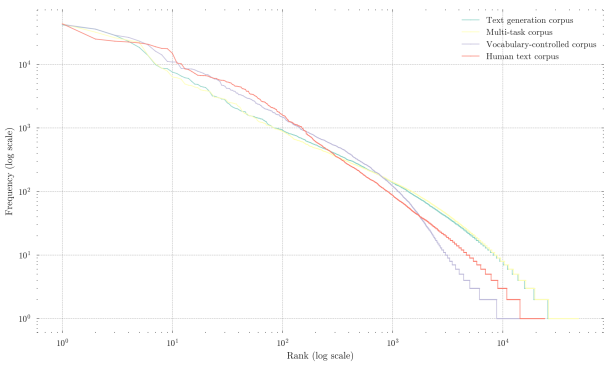

Figure 7: Word frequency in tested corpora.