FunLMs: Methods for Fine-tuning LLMs to Generate Humor

Anonymous ACL submission

Abstract

This paper explores advancements in computational humor through the fine-tuning of large language models (LLMs) on curated datasets of English and Russian jokes. Experiments with fine-tuning on humorous data are conducted to see if generative humor is a viable idea. Different LLMs, sampling techniques and data curation are implemented for enhancing the coherence and effectiveness of humor generation providing a light computational approach for such tasks. We test the proposed methods on different formats and languages and conclude that generating humorous texts with LLMs is feasible though it requires substantial efforts in data preparation and evaluation.

1 Introduction

Computational humor is a significant area of research within natural language processing. The humor analysis is complex due to its reliance on contextual dependencies. Despite this challenge, computational humor offers possibilities for enhancing human-computer interaction.

Previous works in this field (Chen and Soo, 2018; Yang et al., 2015; Mihalcea and Strapparava, 2005) have laid the baseline in humor recognition. Recent research has focused on developing humor generation solutions based on LLMs (Jentzsch and Kersting, 2023; Amin and Burghardt, 2020).

In this study, we introduce new approaches to humor generation task. We also collect, filter and prepare structure-aware humor dataset in English and Russian languages. We fine-tune LLMs on the prepared data, generate new humor samples are evaluate them with human annotators. We also attempt to improve the English model generative abilities by utilizing an additional encoder model that leverages the popular set-up – punchline structure of English jokes. For Russian humor we provide topic-sensitive generation due to its structure features.

The human judgments are obtained for results of all the types of generation approaches presented in this research. They show that various generations techniques might give rise to lightweight humorous models, encourage responsible and effective generative AI. The comparative analysis highlight the significant demand of high-quality filtered data. Unfortunately, though one could cherry-pick some fun and interesting jokes, generative humor remains an open research problem.

2 Related works

Computational humor research has seen significant advancements in recent years, focusing on the understanding, recognition, and generation of humor. One of the primary challenges in this field is capturing the details and contextual dependencies that make humor effective and enjoyable.

Works in computational humor (Mihalcea and Strapparava, 2005; Yang et al., 2015; Chen and Soo, 2018), lay the groundwork by exploring humor recognition field. They introduce linguistic features, which are essential in distinguishing humorous content from non-humorous text.

Humor often involves a delicate balance of positive and negative emotions, the work (Meaney et al., 2021) explores the dual nature of humor and offense, highlighting the importance of sentiment analysis in humor research.

2.1 Set-up – punchline division

The set-up and punchline division is applied in several humor detection systems. For instance, the authors of one study utilize a pretrained language model, **GPT-2**, to calculate uncertainty and surprisal values, which they then use for humor detection (Xie et al., 2021). In another study, the authors compute quantum entropy for the set-up

and punchline, using these values as features in humor detection systems (Liu and Hou, 2023).

2.2 RankGen

The **RankGen** model (Krishna et al., 2022) is a supplementary transformer encoder used to determine if one sequence of words logically follows another within a training text corpus. It is trained using contrastive learning to match the prefix with the correct following sequence and distinguish this prefix from two types of the incorrect (negative) ones. It is used in combination with a language model. Starting with high-quality text generated using LLM and **RankGen** model, text is fed back into this LLM, which produces several possible continuations. **RankGen** selects the best continuation from these options. This cycle repeats until optimal text generation.

This method enhances the quality of the generated text because, while generating text token-by-token LLMs may create incoherent sequences, **RankGen** evaluates and encodes text at the sequence level, leading to more coherent and logical outputs.

3 Data

Data preparation and filtering involved concatenating and filtering jokes in English and Russian. Language-specific steps for each stage of data preprocessing are described in detail in the Appendix A. Table 1 demonstrates features of the resulting datasets.

Data	Number of samples	Features
English	48 336	Division into set-up and punchline
Russian	40 929	Topical prompts associated with each joke

Table 1: Datasets description. In the English dataset, all jokes are divided into the set-up and punchline components. The Russian dataset features a prompt for each joke that contains a topic of the joke. More detailed description can be found in the Appendix A.

Model	Base model	PEFT method
Saiga Mistral 7B	Mistral 7B	LoRA rank 16
Saiga Gemma 9B	Gemma 9B	QLoRA 4-bit
RuAdapt Saiga2 7B	LLaMa-2 7B	LoRA rank 16

Table 2: Description of Russian models and fine-tuning methods.

4 Models

4.1 English LLM

We fine-tune a pretrained **Mistral 7B** (Jiang et al., 2023), a language model with 7 billion parameters, for predicting each token of a punchline given the set-up and previous tokens of the punchline. This allows the model to generate a humorous punchline for a given set-up, token by token. We train the model with and without **LoRA** (Hu et al., 2021) on top of each layer. As expected, increasing the number of training parameters leads to faster convergence speed, see Appendix H for illustrations. For a complete list of experimental parameters, see Appendix D.

4.2 Russian LLMs

We pick several popular fine-tuned Russian models to train on Russian jokes. To save computational resources, we utilize parameter-efficient fine-tuning methods (PEFT) to run each experiment. Table 2 briefly describes the models and methods. All of them had previously been fine-tuned on Russian chat data. One of the models, **RuAdapt** (Tikhomirov and Chernyshev, 2023) features a custom tokenizer developed specifically for the Russian language. We fine-tune each model with a system prompt and without it. The system prompt can be found in the Appendix B.

4.3 RankGen

Additionally, on English jokes we train **RankGen-all-XL** (Krishna et al., 2022), a transformer encoder with 1.2B parameters, to encode the set-up and corresponding punchline into similar vectors, maximizing the dot product of these vectors.

We encode each set-up $\mathbf{s}_i$ from the dataset as prefix and each corresponding punchline $\mathbf{p}_i$ as suffix using the **RankGen** model. We also generate unfunny punchlines $\mathbf{g}_i$ for each set-up from the dataset using the **Mistral 7B** pretrained model and

also encode them as suffixes[1]. Then the objective is to maximize log-likelihood function:

$$\sum_{(s_i,p_i)\in B} \log \hat{P}_\Theta(\mathbf{p}_i|\mathbf{s}_i) = \sum_{(s_i,p_i)\in B} \log \frac{\exp(\mathbf{s}_i \cdot \mathbf{p}_i)}{Z(\mathbf{s}_i)},$$

$$Z(\mathbf{s}_i) = \sum_{p_j \in B} \exp(\mathbf{s}_i \cdot \mathbf{p}_j) + \sum_{g_j \in B} \exp(\mathbf{s}_i \cdot \mathbf{g}_j),$$

where Z is provided for the model trained using the two types of negative examples and different depending on configurations, see details in Appendix F.

For training hyperparameters see Appendix D.

5 Results

5.1 Mistral

To evaluate the quality of the fine-tuned **Mistral 7B** model, we start by taking 100 examples from the test dataset. For these 100 test set-ups, we generate punchlines using our fine-tuned **Mistral 7B + LoRA rank 2** with a sampling temperature of 1.0[2]. We also generate punchlines using our full fine-tuned **Mistral 7B** with the same sampling temperature. For one half of the 100 test set-ups, we generate punchlines using the pretrained **Mistral 7B** model with a sampling temperature of 0.5. For the remaining half, we change the original punchline to another one from the 50 available punchlines.

Each of the 400 jokes is rated by 3 annotators on a scale of one to five (see details about annotation in Appendix E). For each of the five groups presented above, we compute the average humor score, see Table 3.

5.2 RankGen

We calculate different automatic metrics that help evaluate the quality of the resulting **RankGen** models, see Table 4 (formula for scores are in Appendix F).

To evaluate the quality of the best fine-tuned **RankGen** (InBook + Generative) we take 50 test set-ups, generate 100 punchlines for each one using the best fine-tuned **Mistral 7B + LoRA rank 2** model, and then we do the same using **GPT-4o**[3]. After that, for each test set-up we randomly select four generated punchlines: one using **Mistral 7B + LoRA**; one using **Mistral 7B + LoRA with RankGen** as a scoring function; one using **GPT-4o**, and the last one using **GPT-4o with RankGen** as a scoring function.

This way we obtain 200 generated jokes with the same 50 test set-ups, each of which was rated by 3 people on a 5-point scale (see details about annotation in Appendix E). For each class presented above, we calculate the average humor score, see Table 5.

Type	Average score
Dataset	2.87
Mistral 7B (LoRA)	2.7
Mistral 7B (full)	2.68
Mistral 7B (pretrained)	2.58
Dataset (shuffled punchlines)	2.26

Table 3: Average humor scores for each type of jokes. As expected, the jokes with shuffled punchlines are usually nonsense, the jokes with punchlines generated using the pretrained mistral model are typically funnier, the jokes with punchlines generated using the fine-tuned mistral models are funniest among compared models, and original jokes from the dataset are the best.

5.3 Russian models

The models described in the Table 2 are manually evaluated by the authors of this paper. We only tested the models on the prompts they are trained on. Thus, 88 samples have been acquired from each version of the model and it has been found that the models trained with the system prompt generate higher quality text. We have experimented with different values for generation parameters and have chosen a unique set for every model, which are listed in the Appendix D.

Since there is no reliable way to evaluate humor, we annotate the generated Russian samples ourselves. We adopt a three-point scale where we would give a sample a score of 2 if we believe this is a good funny joke, 1 for jokes that are almost funny or seem unfinished or out of context. The rest (unfunny or random text) scores 0. The humor score in the Table 6 represents the sum of those

[1]The original RankGen model encodes prefixes and suffixes into vectors that differ depending on whether the resulting sequence is a prefix or a suffix.

[2]We use ancestral sampling using the temperature of 1.0 since it provides diversity of generated texts without violating the expected output format and generating nonsense. We also try other temperatures and decoding algorithms for LLMs but this one is the best in our opinion.

[3]Our instructions are as follows:
Please, give me 100 funny punchlines for set-up (do not write the set-up in your answers):
<set-up>

Type	Overall score	Relevance score	Humanity score	Funny human-written vs unfunny generative
InBook + Generative	**99.27**	98.71	77.61	89.40
InBook	98.48	**98.83**	43.45	47.20
Generative	78.58	58.45	**86.33**	**92.04**
Reward	87.93	80.05	72.23	79.00
–	88.16	84.37	63.53	62.39
Pretrained	90.73	90.49	53.25	33.37

Table 4: Automatic metrics for **RankGen** models on test data.

Type	Average score
Mistral 7B (LoRA)	2.65
Misral 7B (LoRA) + RankGen	2.69
GPT-4o	3.1
GPT-4o + RankGen	2.87

Table 5: Average humor scores for English generated jokes. RankGen slightly improves the generation quality of the Mistral 7B model trained on the same dataset. Adding the RankGen model also degrades the generation quality of GPT-4o, since GPT-4o was, in our opinion, trained on a higher quality dataset.

points we assigned for divided by the maximum possible score (176 for 88 jokes).

Model	Humor score
Saiga Mistral 7B	0.07
Saiga Gemma 9B	0.16
RuAdapt Saiga2 7B	**0.28**

Table 6: Humor scores for generated Russian jokes.

Some samples of generated jokes can be found in Appendix G.2. Even though training on Russian data did not produce impressive results, we share some observations from our generation experiments:

- The most common artifact in the generated text is repetitions, which for some models have been mitigated with nucleus sampling. Lower values for the top-p parameter lead to reduction in repetitions.
- Introducing even a low repetition penalty (1.1) leads to a significant decline in generation quality.
- Low temperature values (0.1-0.3) cause the model to generate typical sentences (uniform vocabulary) and high values (0.6-0.9) induce hallucination and nonsensical text. This is why we use 0.5 for all models.
- Even though the generated text is not of the highest quality, it is never inappropriate or toxic.

6 Conclusion

In this research, we share the insights we gained while experimenting with fine-tuning models for humor generation.

For the English language, we obtain the lightweight language model, which can be run on a single GPU and, based on a user-specified set-up, is capable of generating a funny punchline closely matching the quality of the dataset. We also test the hypothesis that an additional transformer encoder helps generate better responses since it encodes texts on sequence level while modern large language models do it token by token, potentially leading to lower quality responses. Despite the high values of automatic metrics for the RankGen model, it does not help to improve the quality of generations by capturing the additional structure of the setup-punchline. Comparison of our fine-tuned models with GPT-4o shows the importance of big amount of data, nevertheless, our research provide balanced approach evolving more data depended and computationally effective systems.

It is shown for the Russian language that fine-tuning models with PEFT methods on instruction format data where an instruction contains the topic of the joke can produce some interesting generated funny pieces. The quality of generations could further benefit from more high quality data.

Limitations

The English dataset faces challenges related to the quality and diversity of humor instances. While significant efforts are made to remove offensive and inappropriate content from the datasets, the filtering processes may not be ideal. Enhanced filtering techniques and ongoing monitoring are essential to mitigate this risk. Moreover, future research should focus on developing more advanced generative techniques to enhance the coherence and creativity of the outputs.

When training Russian models, we have only experimented with PEFT methods. Judging by the quality of the resulted generations, the models would benefit greatly from full fine-tuning. Another limitation of this study is the Russian dataset which comprised only ∼40k samples. It seems obvious that collecting a bigger and higher quality humor dataset could further advance humor generation. For example, transcribing popular humor shows seems to be an interesting idea for humor data collection.

Humor is inherently connected to cultural and contextual details, which can be challenging for language models to understand and generate. The models we trained for Engish and Russian show different types of jokes, thus, thwey might not perform well across other cultural contexts and languages, rising the need for culturally adaptive humor generation techniques.

Ethical Statement

The ethical risks that the computational humor research carries are discussed in this section ensure responsible development of AI technologies.

Computational humor systems may mimic biases present in training data. Despite multiple stage filtering process and mitigation strategies to prevent the spread of stereotypes and discriminatory content, generated humorous texts still can sometimes result in offensive or harmful content.

There is a risk that computational humor systems could be misused for malicious purposes, such as cyberbullying, harassment, or spreading misinformation, thus, we claim that our systems are not intended for ant malicious applications.

The rise of computational humor may impact on human writing and entertainment. It is essential to consider societal impacts and strive for a balance that supports human creativity while leveraging technological advancements.

Acknowledgments

We would like to express our gratitude to the volunteer crowdworkers who dedicated their time and effort to evaluate our generative models.

References

Miriam Amin and Manuel Burghardt. 2020. A survey on approaches to computational humor generation. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41, Online. International Committee on Computational Linguistics.

Vladislav Blinov, Valeria Bolotova-Baranova, and Pavel Braslavski. 2019. Large dataset and language model fun-tuning for humor recognition. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4027–4032.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana. Association for Computational Linguistics.

Bhargav Chippada and Shubajit Saha. 2018. Knowledge amalgam: Generating jokes and quotes together. *Preprint*, arXiv:1806.04387.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Sophie Jentzsch and Kristian Kersting. 2023. ChatGPT is fun, but it is not funny! humor is still challenging large language models. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 325–340, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang Liu and Yuexian Hou. 2023. Mining effective features using quantum entropy for humor recognition. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2048–2053, Dubrovnik, Croatia. Association for Computational Linguistics.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4o. https://www.openai.com.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Leonard Tang, Alexander Cai, Steve Li, and Jason Wang. 2022. The naughtyformer: A transformer understands offensive humor. *Preprint*, arXiv:2211.14369.

Mikhail Tikhomirov and Daniil Chernyshev. 2023. Impact of tokenization on llama russian adaptation. *arXiv preprint arXiv:2312.02598*.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3621–3625, Hong Kong, China. Association for Computational Linguistics.

Yubo Xie, Junze Li, and Pearl Pu. 2021. Uncertainty and surprisal jointly deliver the punchline: Exploiting incongruity-based features for humor recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 33–39, Online. Association for Computational Linguistics.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, Lisbon, Portugal. Association for Computational Linguistics.

A Data preparation

For training language models we collected a dataset of English jokes. The existing corpora were not sufficient enough due to small number of instances, high toxicity level, corrupted entries, and big number of semantic duplicates.

We united 16k One-liners (Mihalcea and Strapparava, 2005; Weller and Seppi, 2019; Chen and Soo, 2018), Pun of the Day (Yang et al., 2015), Short Jokes[4], SemEval 2021 Task 7: HaHackathon, Detecting and Rating Humor and Offense (Meaney et al., 2021), Knowledge Amalgam: Generating Jokes and Quotes Together (Chippada and Saha, 2018), The Naughtyformer: A Transformer Understands Offensive Humor (Tang et al., 2022), Humor Detection: A Transformer Gets the Last Laugh (Weller and Seppi, 2019) datasets into one large collection and removed exact duplicates, ignoring punctuation and case. Very short or long entries seem to have a large amount of noise (for example, repetitions, unfunny utterances), thus, we keep examples only between 30 and 150 characters.

The research (Blinov et al., 2019) proposed a dataset of Russian humorous texts.

We have filtered both English and Russian data to remove duplicate, inappropriate and toxic samples using Sentence-BERT (Reimers and Gurevych, 2019), toxicity classifier Detoxify (Hanu and Unitary team, 2020), zero-shot classification model DeBERTa [5]. English entries were divided according to set-up – punchline structure.

The filtering process and data are available in Github repository [6], free from offensive and sensitive content, without duplicates.

We have subsequently clustered this data with k-means algorithms to extract the topics. The number of clusters have empirically been chosen to be 100. We have used the BERTopic framework (Grootendorst, 2022) that assigns the names of the clusters by appending the most representative words to a string.

For English, we have the resulting dataset of 48336 jokes divided into set-up and punchline, which can be used to train language models for generating funny punchlines based on a user-specified set-up. To fine-tune the LLMs, we split the data into train and validation (or test) sets in a 90/10 ratio.

For Russian jokes, the strings reflecting the most representative words for clusters have been then fed to GPT-4o (OpenAI, 2023) with an intention to generate a prompt for LLM (*Напиши шутку про {topic.}* – *Write a joke about {topic.}*). The prompt that we have used can be found in the Appendix B. 88 clusters, subsequently 88 prompts, have been finalized after manual validation. 40929 jokes have been sampled in the final version of the dataset. To fine-tune the LLMs, we split the data into train and validation (20%) sets. Russian jokes do not typically have a setup-punchline structure so we do not divide them and train the models on each joke with a topic-specific prompt that we have generated.

B Prompts

Prompt for GPT-4o to generate a topic-specific prompt for each joke in the Russian dataset:

The following is 100 lines of topics that a dataset of Russian jokes has been clustered into. Write 100 prompts for each topic in the format "Напиши шутку про <topic>." (Write a joke about <topic>.) Infer the topic from the provided cluster names. Be as accurate as possible. If you cannot infer the name of the topic, write "Напиши шутку." (Write a joke.)

System prompt that Russian models have been trained with:

Ты - стендап-комик и сценарист, который с более чем 5-летним опытом занимается созданием юмористических текстов и сценариев для комедийных шоу, а также разработкой концептов для рекламных кампаний и при этом имеет глубокую подготовку в области психологии влияния. Ты нестандартно, интересно и творчески мыслишь. Твоя задача - написать оригинальную шутку с элементом импровизации, опираясь на профессиональные стандарты в области комедии. Ты шутишь только этично и вежливо, никого не оскорбляя. (*You are a stand-up comedian and a screenwriter with more than 5 years of experience who is engaged in creating humorous texts and scripts for comedy shows, as well as developing concepts for advertising campaigns, and at the same time has deep training in the field of influence psychology. You think outside the box, interestingly and creatively. Your task is to write an original joke with an element of improvisation, based on*

[4] https://github.com/amoudgl/short-jokes-dataset

[5] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

[6] https://anonymous.4open.science/r/CleanComedy-6DCA/README.md

professional standards in the field of comedy. You joke only ethically and politely, without offending anyone.)

C Terms of Data Usage

All the data we used in our research is publicly distributed under certain licenses.

1. FUN dataset – MIT License
2. Knowledge Amalgam: Generating Jokes and Quotes Together – MIT License
3. Transformer Gets the Last Laugh – MIT License
4. ShortJokes – Database Contents License (DbCL) v1.0
5. SemEval 2021 task 7 – Creative Commons Attribution 4.0 International License

Our dataset is distributed under Creative Commons Attribution 4.0 International License.

D Hyperparameters

In this section, we provide training and sampling hyperparameters.

To train the whole Mistral 7B model on English data, we use 4 × Nvidia Tesla A100 80GB. For all our other experiments we use a single Nvidia Tesla A100 80GB.

LoRA rank	Epochs	Learning rate
2	2	1e-4
4	2	1e-4
8	2	1e-4
16	2	1e-4
–	1	5e-6

Table 7: Training hyperparameters for Mistral 7B models for English. We use a batch size of 64 and the Adam optimizer with a cosine scheduler and warm-up for the first 3% of training steps.

Parameter	Value
Epochs	5
Learning rate	0.0002
LR scheduler	cosine

Table 8: Training hyperparameters for Russian models.

Type	Learning rate	Epochs
InBook + Generative	1e-4	10.0
InBook	1e-4	11.9
Generative	1e-5	6.8
Reward	1e-6	20.0
–	1e-6	20.0

Table 9: Training hyperparameters for RankGen models. We use the Adam optimizer with a cosine scheduler with warm-up at the first epoch. We train the models for 20 epochs while the loss function on the validation data decreases. We use a batch size of 1024 because such a large size complicates the task of selecting a relevant funny punchline, but makes training the model much more robust and reduces the risk of overfitting. During training we use gradient checkpointing to ensure large batch sizes.

Parameter	Value
Temperature	0.5
Top-p	0.25
Top-k	40
Do Sample	True
Max New Tokens	150
No Repeat Ngram Size	4

Table 10: Generation parameters for Saiga Mistral 7B.

Parameter	Value
Temperature	0.5
Top-p	0.35
Top-k	40
Do Sample	True
Max Length	250
Min New Tokens	20

Table 11: Generation parameters for Saiga Gemma 7B.

Parameter	Value
Temperature	0.5
Top-p	0.8
Top-k	40
Do Sample	True
Max New Tokens	150
Repetition Penalty	1.1

Table 12: Generation parameters for RuAdapt Saiga2 7B.

E Annotation

The annotation was held by crowd-workers volunteered their time, dedicating a maximum of one hour. The scores were collected through

Telegram bot by crowd sourcing, see 1.

> This is a bot that shows jokes both from the dataset we collected and generated using our neural network (there are also bad examples that was combined from different parts of the good jokes)!
>
> You can rate how funny a joke is on a five-point scale, where 1 is not funny, 5 is very funny (do not forget to select also 2-4 scores, choose 1 if the joke is not funny at all!)
> The bot will take your opinion into account that will help evaluate our language model!
> We will be glad if you rate at least 350 jokes (it will not take much time and does not have to be done at once). Of course, if you like it, don't stop after this number! By clicking the "statistics" button, you can find out how many jokes you have already rated.
>
> To better evaluate humor, it is important for us to know your gender, age and language level. We promise that your personal data will be used only in anonymized form to calculate statistics.
>
> Thank you for your time! We really appreciate your desire to improve AI!

Figure 1: Instructions for annotators in Telegram bot.

The text of consent: Before you rate, please read Regulations on Personal Data Processing and confirm your consent.

I confirm that I have personally read Regulations on Personal Data Processing and have the right to provide my personal data and consent to their processing. By providing my personal data, I accept the terms of this Statement, confirm that they apply to me are accurate and up-to-date, thereby freely, of my own free will and in my own interest, dispose of them, understand the consequences of their provision and express my consent to their processing in accordance with clause 3.7 of Regulations on Personal Data Processing.

F Formulae

F.1 Calculating RankGen quality

To compute metrics, we start by building a matrix $A = \{A_{ij}\}_{i,j=1}^{N}$, which shows how well each punchline fits with each set-up in the test dataset according to the RankGen model. Similarly, we build a matrix $B = \{B_{ij}\}_{i,j=1}^{N}$, which shows how well each unfunny punchline, generated using the pretrained **Mistral 7B** model, fits with each set-up in the test dataset. Also, let $C = [A; B]$ represent the concatenation of the matrices A and B. This matrix C has dimensions $N \times 2N$.

Overall score This score reflects the average percentage of preferences of a true punchline from a test dataset to other punchlines from this dataset and generated unfunny punchlines:

$$S_{overall} = \frac{100}{2N^2} \sum_{i=1}^{N} \sum_{j=1}^{2N} \{C_{ii} \geq C_{ij}\}$$

Relevance score This score reflects the average percentage of preferences of a true punchline from a test dataset to other punchlines from this dataset:

$$S_{relevance} = \frac{100}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \{A_{ii} \geq A_{ij}\}$$

Humanity score This score reflects the percentage of preferences of human-written punchlines over machine-generated ones:

$$\hat{C}_i = \underset{j}{argsort}(C_{ij},\ descending = True),$$

$$S_{humanity} = \frac{100}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \{\hat{C}_{ij} \leq N\}$$

Funny human-written vs unfunny generative score This score reflects the percentage of preferences of funny human-written punchlines over corresponding unfunny machine-generated ones:

$$S_{funny\_human} = \frac{100}{N} \sum_{i=1}^{N} \{A_{ii} \geq B_{ii}\}$$

F.2 Loss function for training RankGen

Let $\mathbf{s}_i$, $\mathbf{p}_i$, and $\mathbf{g}_i$ be vector embeddings obtained using the RankGen models and corresponding to the i-th set-up, punchline and generated unfunny punchline, respectively. To train the models we maximize the log-likelihood function

$$\sum_{(s_i,p_i)\in B} \log \hat{P}_\Theta(\mathbf{p}_i|\mathbf{s}_i) = \sum_{(s_i,p_i)\in B} \log \frac{\exp(\mathbf{s}_i \cdot \mathbf{p}_i)}{Z(\mathbf{s}_i)},$$

where Z differs depending on the types of negative punchlines used in training:

InBook + Generative

$$Z(\mathbf{s}_i) = \sum_{p_j\in B} \exp(\mathbf{s}_i \cdot \mathbf{p}_j) + \sum_{g_j\in B} \exp(\mathbf{s}_i \cdot \mathbf{g}_j),$$

InBook

$$Z(\mathbf{s}_i) = \sum_{p_j\in B} \exp(\mathbf{s}_i \cdot \mathbf{p}_j),$$

Generative

$$Z(\mathbf{s}_i) = \sum_{g_j\in B} \exp(\mathbf{s}_i \cdot \mathbf{g}_j),$$

Without negative examples

$$Z(\mathbf{s}_i) = 1,$$

Reward We also train the RankGen model as a reward model that is able to distinguish each funny human-written punchline from corresponding unfunny punchline generated using the Mistral 7B pretrained model.

$$Z(\mathbf{s}_i) = \exp(\mathbf{s}_i \cdot \mathbf{p}_i) + \exp(\mathbf{s}_i \cdot \mathbf{g}_i)$$

The letter B denotes corresponding batch of the dataset.

G Examples of generations

G.1 English

Dataset I'd tell a joke about claustrophobic people, but it might be to tight for you.

What sound does a pedantic owl make? Whom, whom.

Why do sharks like salt water? Because pepper makes them sneeze.

Inbook If aliens are only on the quest for intelligent life, did you say i look like the rock or did you say i look like a rock?

What do you call a scale that always resets itself to zero? to hang with.

What do you call a chicken that can play tennis? The fire department.

Generative I'd tell a joke about claustrophobic people, but that wouldn't be very funny.

Did you hear about the guy who had writers block? He was blocked on writers.

What usually comes after the monster lights the birthday candles? That's right. . . a birthday cake!

Mistral 7B (full) I'd tell a joke about claustrophobic people, but my friends always get the funny bits.

What sound does a pedantic owl make? Too, whoooa.

Why do sharks like salt water? So it won't make them thirsty.

Mistral 7B (LoRA) Why did the calculus students get arrested? Because they had derivatives in their possession.

I'd tell a joke about claustrophobic people, but i don't think there's one that fits right.

What sound does a pedantic owl make? Who cares.

GPT-4o Why did the calculus students get arrested? They were found integrating too many illegal functions.

I bought some toothpaste. And now my ceiling fan has a shiny grin.

What do you call a vampire who owns a restaurant? The Thirsty Table.

GPT-4o + RankGen Why did the calculus students get arrested? They were caught doing improper integrals.

I bought some toothpaste. And now my curtains smell like mint.

What do you call a vampire who owns a restaurant? Chef Bloody Good.

Mistral 7B (LoRA) + RankGen Why did the calculus students get arrested? They were caught derivating.

I bought some toothpaste. My friends told me they were shocked and toothpasted by my purchase.

What do you call a vampire who owns a restaurant? An entremepire.

G.2 Russian

Samples that scored 2 in human evaluation:

Вопрос: Как вам удаётся оставаться молодыми? Ответ: Просто мы ведём себя как будто мы молодые. (*Question: How do you manage to stay young? Answer: We just act like we're young.*)

Шахматы - это игра, где каждый ходит, как ему вздумается, а затем жалуется, когда ему не везет. (*Chess is a game where everyone makes whatever moves they want and then complains when they are unlucky.*)

- Что вы делаете? - Я работаю над проектом по созданию искусственного спутника Земли. - А как это вам удается? - Очень просто. Я взял кусок земли, облепил его вокруг себя, и теперь я сам вращаюсь вокруг него. (- *What are you doing? - I am working on a project to create an artificial satellite of the Earth. - And how do you manage it? - It's very simple. I took a piece of earth, stuck it around me, and now I'm spinning around it myself.*)

Если бы у меня была вторая жизнь, я бы её тоже испортил. (*If I had a second life, I'd ruin it too.*)

Samples that scored 1 in human evaluation:

- Танец начинается, когда мужчина и женщина начинают двигаться друг к другу, а затем неожиданно меняют направление. (- *The dance begins when a man and a woman start moving towards each other, and then suddenly change direction.*)

Вопрос: Как называется место, где люди ведут себя странно, но вежливо?Ответ: Интернет. (*Question: What is the name of the place where people behave strangely but politely?Answer: The Internet.*)

Сон - это прекрасное время, когда ты можешь забыть о своих проблемах и просто наслаждаться сладким сном. (*Sleep is a wonderful time when you can forget about your problems and just enjoy a sweet sleep.*)

Всё как обычно, но вместо Нового года Новый день. (*Everything is as usual, but instead of a New Year, a New day.*)

Samples that scored 0 in human evaluation:

Однажды студент пришел на лекцию по психологии влияния. Профессор начал говорить: "(*One day, a student came to a lecture on the psychology of influence. The professor began to speak: "*)

В этом магазине продаются только те товары, которые покупатели не знают, но нуждаются. (*This store sells only those products that customers do not know, but need.*)

H Figures

In this section, we provide loss function plots for the Mistral 7B model trained on English jokes to demonstrate how the number of trainable parameters affects the number of training epochs required. As observed, the model with LoRA starts to overfit after the second epoch, while the full model starts overfitting right after the first epoch.

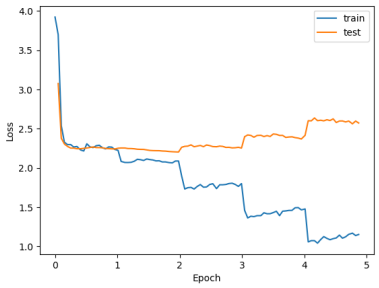

Figure 2: Mistral 7B + LoRA (rank=2)

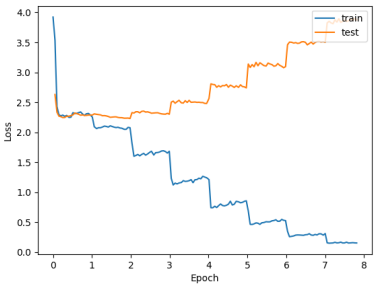

Figure 3: Mistral 7B + LoRA (rank=4)

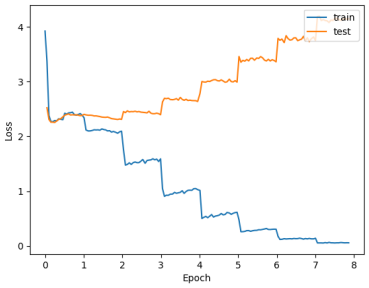

Figure 4: Mistral 7B + LoRA (rank=8)

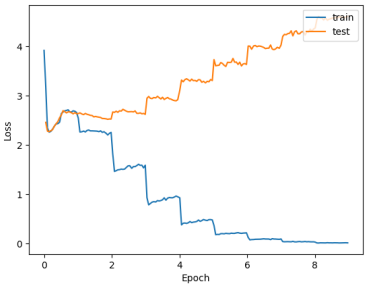

Figure 5: Mistral 7B + LoRA (rank=16)

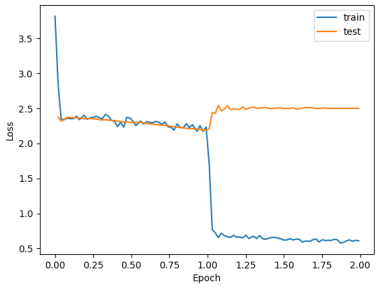

Figure 6: Mistral 7B full fine-tuning (without LoRA)