GPT-4.1-MINI GENERATES SWEDISH NEWS HEADLINES

Project Report: DD2417 Language Engineering, Group 6

Elias Kollberg Gustaf Åberg Léonard Belenge Dalnor

May 2025

Abstract

For this project, we were interested in exploring whether a language model like GPT-4.1-mini can be trained to generate decent headlines for Swedish news articles. Instead of attempting complex architectures, we simply fine-tuned the model on some real article–headline pairs and prompted it to generate a headline for each article. Despite just 50 samples, the model quickly picked up the task and started producing headlines that were coherent, to the point, and surprisingly close to what a human reporter would have written. We evaluated the results both using automatic metrics and our own judgment and found the fine-tuned model to consistently beat the original model. Our study proves that with some targeted practice, it's definitely achievable to get good results, even on a small budget.

- the title of this project was generated by our best model, after reading the abstract above.

1 Introduction

Headlines are what grab your attention first. In a world full of information, they help readers quickly decide if an article is worth their time. But writing a good headline isn't easy, it takes an understanding of the article's message and the ability to express it clearly and concisely. Since this kind of summarization is both important and time-consuming, we wanted to see how well it could be done automatically using a language model.

Our project explores the task of generating headlines automatically from news articles. The idea is pretty straightforward: if you give a model the full article, can it come up with a short, relevant, and well-phrased headline on its own? This falls into the category of abstractive summarization, where the model isn't just copying parts of the article but is actually trying to rephrase the key idea in a new way.

There are lots of summarization models out there, especially encoder-decoder ones like BART and T5, but we decided to go with GPT-4.1-mini instead. It's a decoder-only model from OpenAI, and while it wasn't built specifically for summarization, it's really good at general language tasks. With the right kind of prompt and a bit of fine-tuning, it can learn to generate headlines that feel natural and professional.

Our focus in this project was on Swedish news data, but the general method we used could easily be applied to other languages or topics, as long as you have enough training examples.

2 Background

Generating a good headline is not as easy as it sounds. It's a very compressed kind of summarization, you're trying to capture the main point of an article in just a few words, and it has to be both clear and attention-grabbing. Unlike traditional summaries, which might be a few sentences long, headlines are often just a single line. They need to get to the point fast, and often rephrase things to sound sharp and engaging.

Most modern summarization tools are based on encoder-decoder models. These work by first turning the input text into some kind of internal representation (the encoder), and then generating the output text (the decoder). Models like BART, T5, and mBART have done really well on standard summarization tasks using this approach.

But there's another option: decoder-only models like GPT. We chose to work with GPT-4.1-mini, which is a smaller, cheaper, and more accessible version of OpenAI's GPT-4 series. Even though it wasn't originally designed for summarization, we found that with the right kind of prompting (and a bit of fine-tuning) it could learn to generate headlines just fine.

One nice thing about this setup is that it's simple. We didn't have to build any custom architectures or deal with complicated encoder-decoder setups. We just fine-tuned GPT-4.1-mini on a few dozen examples of Swedish news articles and their original headlines. Each example followed the same format: we gave the model the article text along with a short instruction like "Skapa en rubrik för denna artikel:", and asked it to continue by generating the headline. At test time, we used the same format and let the model do its thing.

3 Data and preprocessing

The dataset used consists of Swedish news articles from 1994 and 1995. It goes over a wide variety of topics, although there may be some bias toward certain subjects. However, this is difficult to determine without looking through all files manually or using a model that can determine topics given a text. The articles vary significantly in length, but the headlines typically contain around 5–8 words. In total, the dataset includes approximately 142,000 articles. Each article is stored in an XML file with various XML tags.

For this project, only the contents within the <HEADLINE> and <BRODTEXT> tags were used, as they contain the news headline and the corresponding article text.

To train the model, the data was converted into JSONL files using the following format:

```
{"messages": [
  {"role": "system", "content": "Du är en assistent som hjälper till att skapa rubriker."},
  {"role": "user", "content": "Skapa en rubrik för denna artikel:<BRODTEXT>"},
  {"role": "assistant", "content": "<HEADLINE>"}
]}
```

This structure is used to properly prompt the GPT model: defining the system's role, specifying the user's request, and demonstrating how the model should respond. For tuning, the GPT-model required only 50 examples for training and validation respectively. And to save on credit costs only 50 examples were used for evaluation.

4 Method and Implementation

To generate headlines, we used OpenAI's GPT-4.1-mini model, a version of GPT designed for efficient inference and accessible through the OpenAI API. Initially, we experimented with zero-shot headline generation by prompting the base model with full news articles and simple instructions like "Write a headline for this article." The results were surprisingly strong, GPT-4.1-mini often produced headlines that were grammatically fluent and topically relevant, thanks to its broad pretraining on multilingual and instruction-following tasks.

However, we quickly noticed a stylistic mismatch. By default, the model would often include multiple suggestions or conversational phrases, e.g., "Sure! Here are a few possible headlines...", which is not desirable in production settings where a clean, single headline is expected.

To fix this, we fine-tuned the model using a small dataset of Swedish article–headline pairs. Each training example consisted of a short article followed by a prompt such as:

```
Skapa en rubrik för denna artikel:
[ARTICLE TEXT]
```

The target was the correct headline, written concisely and without additional formatting. This setup encouraged the model to generate direct, newspaper-style headlines, short, capitalized, and focused. For example, after fine-tuning, the model produced outputs like:

```
TORGNY MOGREN FICK JERRINGPRISET
```

We fine-tuned the model via OpenAI's API, using only 50 examples (recommendation in the finetuning portal). Despite the small size, this was enough to shift the model's output style significantly. All generations during evaluation were performed with a temperature of 0.0 to ensure deterministic output. We did not alter other sampling parameters.

4.1 Hyperparameter Search

Since our dataset was quite small, we ran a small grid search to find a training configuration that would help the model adapt efficiently without overfitting. We experimented with combinations of batch size, learning rate multiplier, and number of training epochs. More specifically, we tested batch sizes of 2 and 4, learning rate multipliers of 2 and 3, and training for either 2 or 3 epochs. This resulted in 8 configurations in total.

We performed the search manually using OpenAI's fine-tuning API, and we evaluated each run based on how well the outputs of the model matched the style and content of real headlines. An aside here is that we had originally wanted to perform a larger grid search over more parameters. However, since each fine-tuning run through OpenAI's API costs actual money, and our project was on a budget, we opted to keep the search small and focused. Even so, even our limited tuning was enough to improve headlines significantly, and we suspect further optimization would yield even more gain.

5 Evaluation

To check how well our fine-tuned model performs at generating headlines, we ran a small-scale evaluation on 50 examples. Each example consisted of a Swedish news article and a gold-standard headline written by a human. These examples were not seen during training and represent a mix of all kinds of topics, but mainly sports and general news.

5.1 How We Evaluated

We used a combination of some more mathematical metrics along with our own manual inspection to assess headline quality. For the more theoretical metrics, we focused on these three:

- **Exact Match:** We checked whether the model-generated headline **exactly** matched the gold headline, except that we ignored the differences in capitalization. This is a strict measure which we do not expect to get a good sucess rate, but it helps us count how often the model perfectly reproduces the target.
- **String Similarity:** To get a more flexible sense of similarity, we used Python's `SequenceMatcher` to compute a character-level similarity score between the generated and gold headlines. This produces a value between 0 and 1, capturing how close the strings are (even if they aren't identical).
- **ROUGE:** Finally, ROUGE scores were used to evaluate how much n-gram overlap there was between the generated and gold headlines. Both ROUGE-2 (bigram) and ROUGE-3 (trigram) scores were evaluated. Let the following be defined:
  - R_{bigrams} = set of bigrams in the reference
  - P_{bigrams} = set of bigrams in the prediction
  - overlap = $R_{\text{bigrams}} \cap P_{\text{bigrams}}$

  Then the ROUGE metrics are defined as:

$$\text{Recall} = \frac{|\text{overlap}|}{|R_{\text{bigrams}}|}$$

$$\text{Precision} = \frac{|\text{overlap}|}{|P_{\text{bigrams}}|}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- **Recall** measures how much of the reference bigrams are captured by the prediction.
- **Precision** measures how many of the predicted bigrams are correct.
- **F1 Score** is the harmonic mean of precision and recall.

And the F1 score is the given output score used for evaluation, where a score closer to 1 means better performance.

To generate predictions, we used OpenAI's API to query our fine-tuned GPT-4.1-mini model on each example. We also set the temperature to 0.0 to ensure deterministic output and avoid variability in between runs.

5.2 What We Looked For

In addition to the quantitative metrics, we manually reviewed all predictions which allowed us to evaluate aspects that are harder to capture with numbers. Things we might have checked are:

- Does the headline capture the main idea of the article?
- Is it grammatically correct and fluent?
- Does it look and feel like a real Swedish news headline?

Overall, the model performed quite well. Even when it didn't produce an exact match, it generated headlines that were just as informative or even stylistically preferable. The outputs were generally short, and aligned with journalistic style. There were some weaker cases, for example, when the model focused on a side detail or left out a key name, but these were relatively rare. We present the full results in the next section.

6 Results

Model	Batch Size	LR Multiplier	Epochs
1	2	3	2
2	2	3	3
3	2	2	2
4	2	2	3
5	4	3	2
6	4	3	3
7	4	2	2
8	4	2	3

Table 1: Hyperparameter configurations used for each model.

With the hyperparameter configurations defined in Table 1, the evaluation results for each model are presented in Table 2. These include training and validation losses, as well as performance metrics such as exact match rate, test accuracy, average string similarity, and ROUGE scores.

Model	Train Loss	Val Loss	Exact Matches	Test Accuracy	Avg Similarity	ROUGE-2	ROUGE-3
1	0.933	0.684	7	14%	52.30%	26.84%	13.46%
2	0.243	1.480	8	16%	53.59%	26.00%	12.18%
3	1.071	0.891	5	10%	52.29%	22.68%	15.28%
4	0.320	0.318	8	16%	53.25%	26.56%	14.29%
5	0.645	1.209	4	8%	51.60%	24.85%	14.54%
6	0.487	1.370	6	12%	52.44%	24.04%	10.30%
7	0.781	1.159	2	4%	50.03%	21.50%	13.78%
8	0.570	1.379	5	10%	49.19%	21.94%	12.03%
Non tuned GPT model	-	-	0	0%	32.74%	10.48%	4.89%

Table 2: Performance of different hyperparameter configurations on headline generation task.

Based on the combined performance across all metrics, Model 4 was the best-performing configuration. It got some pretty strong scores in exact matches (8 out of 50), test accuracy (16%), average similarity (53.25%), and ROUGE metrics, while also having the lowest validation loss (0.318). As can be seen in Table 2 the non-tuned GPT model performed significantly worse than the tuned model, with a 20% less accuracy and achieving only around half the ROUGE scores. The loss curve for this model looks as follows:

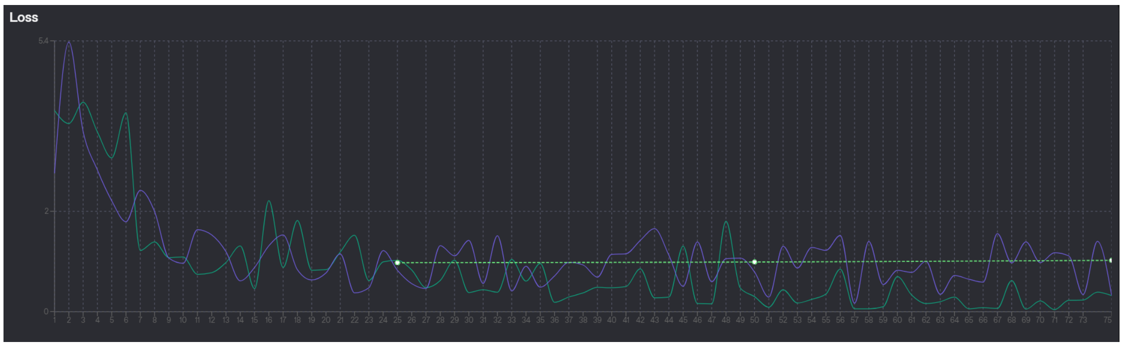

Figure 1: Loss curve for model number 4, (purple: validation loss , green: training loss)

While overall performance was pretty strong, the model did not always succeed in generating a headline that matched the reference. In some cases, there was a clear mismatch between the generated and gold headlines, though the output was still reasonable from a journalistic point of view.

One such example is shown below:

```
--------------------------------------------------
Example 17
Prompt: Skapa en rubrik för denna artikel:

Avd I: Gale Warning (6) ...
Ideal: DAGENS DUBBEL (TÄBY)
Predicted: VÄRMLANDS STORA LOPP - 1. GALE WARNING, 2. FISHERMAN'S GIRL, 3. SISTER ALICA
Exact match: False
Similarity: 0.10
ROUGE-2 F1: 0.00% | ROUGE-3 F1: 0.00%
--------------------------------------------------
```

We noticed that the fine-tuned model had clearly been trained on news data from 1994 and 1995. Interestingly, all the titles in the dataset were written in capital letters, even though that might not have been the case in the original newspapers. This suggests that our model has learned to closely mimic the style of the dataset. Headlines often started with the category followed by the actual headline, for example: "SPORT: SÅ GICK DET I HELGENS MATCH". Consequently, the fine-tuned model consistently produces headlines in this old-fashioned style, which feels a bit unusual, as news headlines in Swedish today typically don't follow this format anymore. This is demonstrated below.

```
--------------------------------------------------

Example 40
Prompt: Skapa en rubrik för denna artikel:

SIMVM: MEDALJSTÄLLNING: ...
Ideal: SIMVM: MEDALJSTÄLLNING: SVERIGE TVÅA I MEDALJSTÄLLINGEN
Predicted: SIMVM: MEDALJSTÄLLNING: SVERIGE TVÅA I MEDALJSTÄLLINGEN
Exact match: True
Similarity: 1.00
ROUGE-2 F1: 100.00% | ROUGE-3 F1: 100.00%
--------------------------------------------------
```

Although the predicted headline scores poorly in terms of similarity and ROUGE, it is not necessarily wrong. It highlights race results which is a reasonable alternative focus. This illustrates an important limitation of automated evaluation: headline generation is not a deterministic task, and different editors might write different (yet equally valid) headlines for the same article.

7 Discussion

When we first tried to produce headlines using GPT-4.1-mini without fine-tuning, we were amazed at how well it was performing initially. Just by instructing it to "write a headline for this article," it could generally come up with something that was grammatically correct and kind of on topic. But the tone was generally way off. Instead of delivering a tidy, newspaper-style headline, the model would sometimes give multiple suggestions or say it in the voice of a chat, e.g., "Sure! Here are some headline suggestions."

That kind of output may be fine in conversation, but it's not what you're looking for if you want to incorporate the model into something that just has to have a single tidy headline. Thus, we trained the model on real-life examples, pairing short Swedish news articles with the titles they originally had. With just 50 training examples, the contrast in output style was noticeable. The model started producing far more realistic, snappy, and concise headlines, exactly what we were after.

Of course, it wasn't perfect. There were still some places where the model picked up on small things, or where its response didn't fit the tone or topic of the human-created headline. But these were exceptions. Most of the time, even if the headline wasn't a carbon copy, it worked. It would typically pick up on the gist and convey it.

From these results, Model 4 stood out. It had the lowest validation loss tied with the highest exact number of matches, and it had high ROUGE scores. All of our fine-tuned models were better than the non-fine-tuned model, in a couple of cases, a lot better, showing that even a small amount of focused training can have a large impact.

One thing we realized along the way is that judging headlines automatically is not without its restrictions. A headline could be "wrong" in ROUGE or string similarity, but downright reasonable. In our horse race example, the gold headline was focused on a novel event name, while the model provided the race outcome instead. Both are good summaries, they're just different in their approach.

Another thing we discovered towards the end of this project was that all the headlines in the dataset were written in capital letters. This didn't seem reasonable to us, based on what we had seen in modern news articles. We even tried to find old Swedish newspapers from 1994-95 to see if they were always written in capital letters back then, but we couldn't find any evidence of that. We don't know why the dataset was formatted this way, perhaps it was done for simplicity. If we were to redo this project more thoroughly, we would have considered this issue earlier and done something about it, such as preprocessing the data to remove the categories from the headlines and to fix the capitalization. Even though this formatting makes the fine-tuned model less representative of how headlines are generally written, it still demonstrates that a fine-tuned LLM can adapt very well to its training data and produce consistent outputs tailored to a specific style.

In the future, we'd love to explore how much further this approach could be pushed with additional data and a wider hyperparameter search. But even under resource constraints, the outcomes looked promising.

8 Conclusion

For this project, we wondered if GPT-4.1-mini could be used to generate headlines for news headlines, and more specifically, if it would get better if we fine-tuned it on real examples. The base model already generated pretty good results straight out of the package, but after fine-tuning it on a small set of Swedish article–headline pairs, quality improved noticeably. The headlines became crisper, more blunt, and read a lot more like something that an actual reporter would actually write.

The method that we used was really straightforward. We did not have big infrastructure or big data, but despite some constraints, we were able to train a system that worked well. That's one of the strengths of working with GPT-based models, if you have good prompts and some tweaking, you can achieve really excellent results on a very specific task.

All things considered, however, there is definitely potential for improvement. Because we had budget constraints, we couldn't perform a big hyperparameter search or try more advanced tuning strategies. With more time (and money), we might get even stronger results by trying out other configurations or training on more data.

However, the results speak for themselves. Even slight amounts of targeted fine-tuning produced a noticeable difference, which should suggest that this kind of configuration could be useful to implement in other situations too, for instance, summarizing blog articles, title generation for research papers, or translating the model to other languages.

References

- OpenAI. (2024). *GPT-4 Technical Report.* Retrieved from `https://openai.com/research/gpt-4`
- OpenAI. (2024). *Fine-tuning GPT-4 and GPT-3.5 Models.* Retrieved from `https://platform.openai.com/docs/guides/fine-tuning`
- clefkorpus (1994-1995). *Swedish newspaper dataset.* Retrieved from `https://kth-my.sharepoint.com/personal/jboye_ug_kth_se/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fjboye%5Fug%5Fkth%5Fse%2FDocuments%2Fdatasets%2Fttclef%2Fttclefkorpus%2Etar&parent=%2Fpersonal%2Fjboye%5Fug%5Fkth%5Fse%2FDocuments%2Fdatasets%2Fttclef&ga=1`