Assignment: Generating Weather Reports with LLMs 🌦️

Outcomes and takeaways

Zdeněk Kasner

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

☀ Task #1: Generate the current weather description

1a) Do the reports look the way you would expect?

- short reports are (pretty much) fluent and accurate on the first sight, but there are some issues:
  - irrelevant data
  - occasional weird phrasing, copying values exactly
  - US units (°F) for US cities
  - mishandling Unix-time timestamps
    - sunrise time 1711601215 → “The sun is currently rising at 17:11:43 UTC”

- default decoding parameters do not work for all the models

☀ Task #1: Generate the current weather description

1b) How can you improve the results with a better prompt?

- techniques:
  - preprocessing the data (simplifying / transforming JSON, converting units and times)
  - asking to present only weather-related data
  - asking to present a stylized report (e.g., “as in the radio”, tabular format)

- does not work for all the models equally
  - for the base model, it helps to prepend “The weather forecast for”

- providing an example hurts, the model tends to repeat parts of the example

Task #1: Generate the current weather description

1c) How can you improve the results by varying decoding algorithms (beam search, top-k, top-p, ...) and their parameters?

- both greedy decoding sampling with low k / p work reasonably well
- lowering the temperature is ok
- high temperature gives unexpected results
- using more beams leads to unnecessary repetitions

- overall, the choice of the decoding parameters can have a big impact on the result quality (although some models are more robust)

☀️ Task #1: Generate the current weather description

1d) What differences between the models do you observe?

- [node 1] Mistral-instruct
  - very consistent and usable
- [node 2] Mistral-base
  - unusable in default setting (empty outputs etc.) → needs more thoughtful prompting
- [node 3] Phi-2
  - works, but sensitive to prompt and parameters, extra outputs
- [node 4] Aya
  - short and simplistic answers

☀ Task #1: Generate the current weather description

> *We tried appending “but round the temperature values to whole numbers and make it sound fun”. The model made it sound fun, but it was not great at rounding the numbers*

> *asking it to always use units can bring missing units back, but can also lead to “200-year-old Cod” and other oddities*

> *add Let's think step by step. to the end of the prompt*
→ *no improvement*
→ *this task does not need thinking step by step*

☀ Task #1: Generate the current weather description

> I prefer sampling over greedy decoding for instruction-tuned Mistral, however CohereForAI might start generating Sci-Fi stories when it is enabled.

> Aya seems to ignore the prompt engineering, modifying the hyperparameters doesn't help with that and activly degreades the output quality.

> Model at node 3 would randomly give python programming exercises in the weather reports.

Task #2: Generate a 5-day forecast

2a) Which qualities would you expect the weather forecast to have (i.e., how should the generated text be evaluated)?

- **consistent:** following a logical flow / structured
- **concise**: without superfluous details
- **complete:** not withholding important details
- **correct:** accurately reflecting the data

> it should clearly state that it is just a prediction

Task #2: Generate a 5-day forecast

2b) Do the generated reports have these qualities? If not, what are the issues?

- mostly not
  - inaccuracies, extra fields, too concise
  - consistency may vary
- depends on the decoding parameters, prompt, and model
- additional problems with context size →

Task #2: Generate a 5-day forecast

2c) How does filtering data items or removing certain fields from the input data improve the output quality?

- unequal context size for each model → different pruning factor needed
- for long inputs, the initial prompt will not be seen by the model
  - behavior of the model when “instructed” by an incomplete JSON is undefined (Phi-2 apparently starts to spit out random Python code)
- pruning is not ideal: *“When we set the pruning factor to 8, we get weather forecasts for the next 5 days, for each only one data-point. This approach is effective, however **not very correct,** because of the timestamps. If we just prune all of the forecasts for the cities in the same way, we get forecasts **for the US cities in the early morning, for Europe in the afternoon and for Asia late at evening or night**.”*

Task #2: Generate a 5-day forecast

2d) Do the insights from 1b)-1d) apply for this task as well?

- generally yes
- Mistral shines even more thanks to its longs context size (8k)

> All previous results are all from node 1 (Mistral-7B-instruct) and it performed the best. Node 2 (Mistral-7B-base) continues writing the instructions about how the output should look like. Node 3: outputs the prompt converted back to json (it was given a yaml prompt). Node 4: Just outputs the word "Amsterdam"

Task #2: Generate a 5-day forecast

> *converting to yaml decreases the token count by almost 23%.*
> *https://tiktokenizer.vercel.app/?encoder=cl100k_base*

```yaml
cod: '200'
message: 0
cnt: 40
list:
- dt: 1711638000
  main:
    temp: 23.22
    feels_like: 23.66
    temp_min: 23.22
    temp_max: 24.24
    pressure: 1018
    sea_level: 1018
    grnd_level: 824
    humidity: 79
    temp_kf: -1.02
```

```json
{
    "cod": "200",
    "message": 0,
    "cnt": 40,
    "list": [
        {
            "dt": 1711638000,
            "main": {
                "temp": 23.22,
                "feels_like": 23.66,
                "temp_min": 23.22,
                "temp_max": 24.24,
                "pressure": 1018,
                "sea_level": 1018,
                "grnd_level": 824,
                "humidity": 79,
                "temp_kf": -1.02
```

🇺🇳 Task #3: Generate a weather report in another language

3a) Do you observe a drop in quality compared to English? If yes, what problems do you observe?

- languages tried: German, Russian, Czech, Italian, Spanish, Persian, Slovak, Macedonian, Serbian, Hebrew
- models sometimes have troubles sticking to the language
- qualitative changes:
  - in Italian, the report tends to be more structured
  - non-sensical outputs for Persian
- differences between the models →

🇺🇳 Task #3: Generate a weather report in another language

3b) How do the models compare on this task?

- Aya performs the best for many languages (Russian, Czech, also the only model working for Persian)
- some people say that models abilities did not change, preferring Mistral
- however, Mistral in Czech *"uses a lot of misspelled words, wrong word order, wrong inflections"*

- **Mistral:** "V Amsterdamu je dnes 11,22 °C a cítí se 10,17 °C. Temperatura se bude pohybovat mezi 10,47 °C a 12,22 °C. Větr je s rychlostí 10,28 m/s a směrem 207°. Vlhkost je 68 % a vizibilita je 10 000 m. Vlaky jsou rozptřené s 40 % obsahu."
- **Aya:** "V Amsterdamu je dnes oblačno s teplotou 11°C a větrem o rychlosti 10,28 km/h. V noci bude jasno s teplotou 10°C a větrem o rychlosti 13,41 km/h."

🇺🇳 Task #3: Generate a weather report in another language

3c) What would be your method of choice for generating weather forecasts in this language in a practical scenario?

- suggestions:
  - generate reports in English → translate
  - use a multilingual model, prompt in the target language
    - pre-translate the data record (along with city names etc.) to help the model
  - finetuning on the target language
  - GPT-4 API
  - simple template-based approach (filling out blanks in a prepared text)
  - human being

🇺🇳 Task #3: Generate a weather report in another language

> [de] There is no noticeable drop in quality for Microsoft-Phi's code or Base-Mistral's silence.

> [ru] "Amsterdam" being translated to "Moscow" in Russian outputs

> [it] The fourth model (...) adds the feature "Gusto del vento" ("Taste of the wind") which is 13.42 °C.

> [sk] First, we tried writing the prompt in English and specifying the language in the prompt. This resulted in a text that was half in English and half in Slovak. (contd.)

🇺🇳 Task #3: Generate a weather report in another language

(...)
The prompt that worked the best was in Slovak and explicitely specified the language in the prompt. However, the output we received was a combination of all kinds of slavic languages especially Czech, Croatian and Slovak. A funny example is:
Dobro jutro, dobro zjutro!
Dnes v Amsterdamu bude být den s oblačností

> The model also fails on some very low resource languages like elf language and emojis.

Task #4: Generate stylized weather reports

4a) How do you need to modify the prompt to generate stylized reports?

- stylizations tried: bullet-points, rhymes, song-lyrics, irony / sarcasm, personalized (e.g. for cyclists, for kids), Shakespearean style, emojis, one-sentence length, markdown table

- stating the request in the prompt explicitly seems to be often enough

Task #4: Generate stylized weather reports

4b) Are the responses robustly following the style for every output? If not, can you make it more robust?

- increasing top-k (and temperature) helps for more creative outputs, but may harm consistency
- stylization may vanish after the first sentence
- models:
  - Mistral much better than Phi-2 and Aya

Task #4: Generate stylized weather reports

4c) Is there a difference in quality of the outputs compared to the default setup?

- stylization may actually help the fluency
- however, the output may be less thorough and precise

> This report reads well, so its length is not a problem. However it is too optimistic, I wouldn't wind over 10 m/s gentle (it is actually twice the speed of gentle breeze on the Beaufort scale).

Task #4: Generate stylized weather reports

> In Prague, the day is fair and bright,
With a clear sky and no clouds in sight.
The temperature is mild, at 14.88 degrees,
And feels like 13.31, with no need for degrees.
The wind is light, at 9.26 miles per hour,
Coming from the west, with a gentle power.
The pressure is at 990, with humidity of 34,
And visibility of 10,000 meters, clear and clear.
The sun is rising at 1711601215,
And setting at 1711646845, a beautiful sight to see.
So come out and enjoy the day, it's a perfect time to play.

Task #4: Generate stylized weather reports

> Once upon a time, in the magical city of Amsterdam, the sun was shining bright and the sky was a beautiful shade of blue. But then, the clouds started to gather and the sky turned a bit grey. The wind started to blow a little bit stronger and the temperature dropped a bit.

Task #4: Generate stylized weather reports

> In the Atlanta, where Atlanta, our scene, theoth rise and set, a most splendid day hath beenacedaced upon us. The Theies are clear as crystal, with nary a cloud in sight to mar the the of the heavens. The temperature doth stand at a brisk 9.98 degrees, with a feel of 7.166, making it aoth blow gently from the west, with a speed of 6.177 per hour and a gust of 11.32. The humidity doth stand at stand at6%,6%, air the air most breathe. breathe. doth stretchoth stretch ten thousand ten thousand, allowing allowing us to see all that lies before us. The sun doth rise at 5:37 am and set at 7:01 pm, casting its golden gloways upon the... pressure doth stand at 1017 hPa, and the city doth standrive under the watchful eye of the United States. The in all, a most glorious day hath been bestowed upon Atlanta. and we must noblest of ofations.

Task #4: Generate stylized weather reports

“One sentence”:

> Currently in Amsterdam, it is cloudy with scattered clouds and a temperature of 11.22°C, feeling like 10.17°C with a pressure of 983 hPa and humidity of 68%. Wind is blowing at 10.28 m/s from the southwest with gusts up to 13.41 m/s. The visibility is 10,000 meters and the all-cloud coverage is 40%.

General feedback on reports

- reports were nice and thorough, thanks! 🙏
- most people agreed on the outcomes
  - but it may vary by selected language and style
  - also by the default set of decoding parameters people were using
- one team may have used another LLM to compare the LLM outputs 😜

- open LLMs are promising, but seemingly have still a lot of space for improvement