AI on Track: Generating Accurate and Engaging Athletics Articles with Language Models

Finn Alberts (852685751) [1] **Ewoud Vosse (852644258)** [1]

1. Brief Introduction

Since the groundbreaking introduction of the Transformer architecture by Vaswani et al. (2017), the development of Large Language Models (LLMs) has seen an immense growth. Before Transformers, language generation, which is a form of sequence modelling, was mainly done using recurrent neural networks and Long Short-Term memory. Transformers, however, significantly improved both output quality and performance. The main idea of the Transformer architecture is self-attention, where the input is processed by an encoder which enriches it by creating connections between words in the input sequence. This allows for a much better understanding of a given input.

The paper of Vaswani et al. (2017) can be considered a breakthrough and led to the development of many of the now famous LLMs, including BERT (Kenton & Toutanova, 2019), GPT (Radford, 2018), GPT2 (Radford et al., 2019) and GPT3 (Brown, 2020). Especially the GPT models became well-known, because of their accessibility through the ChatGPT interface, which also allowed anyone to experiment with it, even those who do not have a technical background. This led to the development of many LLM-based applications, such as customer service chatbots.

As LLMs have proven versatile across domains, one interesting application area is in copywriting—particularly in creating articles summarizing sports events. In this project, we will focus on trying to automate the process of writing the articles for athletics events, including both track and field events as well as running events. When writing these articles, we are looking for a balance between factual correctness, as no incorrect results should be included in the final output, as well as readability, because these articles should be enjoyable to read and should thus not be a simple recital of the results. We will be using Scopias Atletiek, a local athletics club from Venlo, The Netherlands, as an example for the duration of this project. The results, however, can be generalised for other clubs as well. We divide up this project in three research questions:

1. What is the accuracy and factual reliability of an LLM in generating summaries of track-and-field and running event results for athletes from Scopias Atletiek?
2. How does the format of input data (such as CSV files, email summaries, or result webpages) influence the articles generated by an LLM?
3. How do varying prompting techniques affect the articles generated by an LLM?

In the next section, section 2, the research method is described, including the data (2.1) and the prompting strategies (2.2). In section 3 the results of the experiments are given, followed by the discussion in section 4 and at last the conclusion in section 5.

2. Methods

In recent years, various frameworks have been developed to test and evaluate the performance of LLMs. The research of Arawjo et al. (2024) focused on a simple and easy to use UI to compare the performance of different LLMs. Another framework is BADGE, the work of Chiang et al. (2024), a framework that creates reports of Badminton matches. Our research continues on the work of Chiang et al. (2024), to expand the framework to also be applicable to athletic matches.

2.1. Datasets

LLMs are very sensitive to how the input is structured, given the countless papers on this topic. Although the results from the BADGE study (Chiang et al., 2024) do not show a big difference between using data from a CSV file or from the Q&A method, their data was more structured.

In this study, the data input is more varied and therefore categorised into:

- Structured (CSV): information including, but not limited to, athlete name, result and ranking in a structured way
- Unstructured (HTML): information like athlete names, match type and result, scraped from Atletiek.nu and Uitslagen.nl, and cleaned using BeautifulSoup[1].

[1] https://beautiful-soup-4.readthedocs.io/en/latest/

- Raw text (trainer notes): notes from various trainers with different types of information, depending on the trainer, but including athlete name from Scopias, match type and result.

For every type of input, 5 training examples are prepared. Examples of these inputs can be found in Appendix B.

2.2. Prompting Techniques

Similar to (Chiang et al., 2024), this study also uses five prompting techniques: Zero-shot, One-shot, Few-shot, Chain of Thought (CoT) and Auto Chain of Thought (Auto CoT). Those five prompting techniques differ in how and how many examples are given (DAIR.AI, 2024):

- Zero-shot: no examples are given.
- One-shot: only one example is given.
- Few-shot: multiple examples are given.
- Chain of Thought (CoT): zero or more examples can be given (in this study we provide zero), but the instruction is given by demonstrating step by step, to avoid skipping to a wrong answer.
- Auto Chain of Thought (Auto CoT): similar to Chain of Thought but the model does not require step by step examples, to execute Chain of Thought reasoning, as it will generate these steps itself.

The examples required for One-shot and Few-shot are acquired from www.scopias.nl and translated into English using Google Translate.

2.3. LLM Selection

There are many metrics that define the performance of an LLM. For our model selection, we specifically look for a model which performs well for the IFEval metric (Zhou et al., 2023). This metric is designed to measure how well a model follows given instructions. This is especially useful for our use case, as we want our model to use the input data well and extract the results properly. Other metrics, such as Big Bench Hard (BBH) (Suzgun et al., 2022), MATH (Hendrycks et al., 2021), Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023), Multistep Soft Reasoning (MuSR) (Sprague et al., 2024) or Massive Multitask Language Understanding - Professional (MMLU-PRO) (Wang et al., 2024) are more focused on assessing the model's logical reasoning abilities, factual recall, and problem-solving skills in structured, knowledge-based tasks. Hugging Face, a major website which hosts a lot of LLMs and datasets, offers a leaderboard[2] where one can get an overview of these metrics for their models. We first apply some filters, as we only want chat models and/or pretrained models. We do not want to limit the amount of parameters. Then, we pick the model with the highest IFEval.

The best available model is the Meta Llama 3.1 instruct model with 70B parameters with a IFEval score of 86.69 (LLama Team, AI @ Meta, 2024). The second best option is Calme-2.1-Qwen (IFEval of 86.62), a finetuned version of Qwen 2.5 with 72B parameters. The third best option is the non-finetuned version of Qwen 2.5 (IFEval 86.38) (Yang et al., 2024; Qwen Team, 2024). Given our access limitations with Llama and Calme models (see more in the next section), the best model of choice is Qwen 2.5 with 72B parameters.

2.4. Implementation Details

For the implementation of the prompt engineering techniques, we used the Hugging Face Serverless Inference API, due to local hardware limitations. This free tier of this API allows users to make 1000 requests/day for models up to 10 GB in size. Therefore, we could not pick the best available model and had to settle for the Qwen 2.5 model.

The process of generating an article consists of two steps. First, we build up the prompt using the `prompt_builder` function. This function takes the prompt engineering technique and the input data and builds the prompt accordingly. For the One-shot technique, we randomly select one examples from the set of available examples. For the Few-shot technique, we select three examples. When Auto CoT is picked as the prompt engineering technique, the function makes a call to the `generate_text` function to generate the chain of thought. The `generate_text` function makes the calls to the Hugging Face API. For regular CoT, we include a standard chain of thought.

After the prompt has been built, the prompt is sent to the Hugging Face API using the `generate_text` function. This means that for Auto CoT two API calls are made, while for all other techniques only one call is made. Also see Figure 1 for an overview of the main flow of the implementation.

The generating of the chain of thought (for Auto CoT) and the generating of the article itself both have their own prompts. These prompts can be found in Appendix A.

[2] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

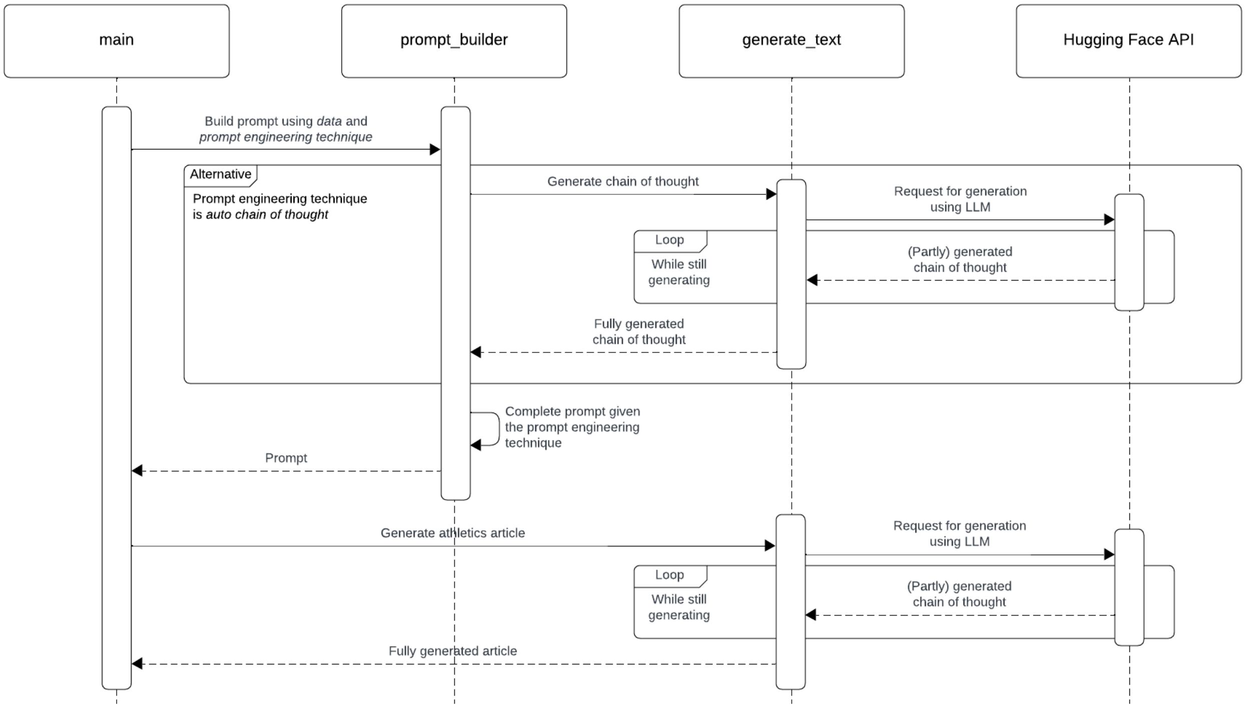

main
prompt_builder
generate_text
Hugging Face API
Build prompt using *data* and *prompt engineering technique*
Alternative
Prompt engineering technique is *auto chain of thought*
Generate chain of thought
Request for generation using LLM
Loop
While still generating
(Partly) generated chain of thought
Fully generated chain of thought
Complete prompt given the prompt engineering technique
Prompt
Generate athletics article
Request for generation using LLM
Loop
While still generating
(Partly) generated chain of thought
Fully generated article

Figure 1. Main flow of the application showing how different functions interact with each other

2.5. Experiment Setup

This experiment consists of 3 types of inputs and 5 different prompting techniques, leading to an experiment with 15 different parameters. For every parameter, 5 example articles are generated. The five articles are evaluated and the average is taken as a final evaluation score of every parameter, as shown in the table 1.

	CSV	HTML	Notes
Zero-shot			
One-shot		5 articles	
Few-shot		and 1 average score	
Chain of Thought		per parameter	
Auto Chain of Thought			

Table 1. Experiment Parameters

2.6. Performance Evaluation

In their research, Chiang et al. (2024) used GPT4 to evaluate the articles on four aspects: Coherence, Consistency, Excitement and Fluency. As one of the models used to write the report, was also GPT4, they admitted in their conclusion that there might be bias in evaluating the articles. Arawjo et al. (2024) developed a model to evaluate models by means of an UI, giving humans the opportunity to evaluate the answers from different models from given prompts. Both evaluation methods, evaluation by a model or evaluation by a human (by researchers or participants), have pros and cons. Due to the time constraint factor of this study, the evaluation model chosen is human evaluation by the researchers. Although the bias from the researchers can form a problem, one has extensive knowledge about athletics and the other does not, balancing existing bias about the contents of the article. In Table 2 the criteria are shown, where every criterium is given a score from 0 to 2.

Criteria	**Scores**		
	0	**1**	**2**
Athletes mentioned	None	Some	All
Factually correct content	Too little	Mostly	All
Fluent article	No	A bit	Yes
Exciting to read	No	A bit	Yes
Confabulations	A lot	Some	None

Table 2. Evaluation Criteria

To clarify the criteria a bit further, ”Athletes mentioned” is to make sure that all athletes are mentioned and receive the attention that they deserve. The ”Factually correct content” criterium is to make sure that results are not manipulated and ”Fluent article” to ensure that an article with sufficient quality is produced by the model, meaning it should have a structure that is to be expected for an athletics article. ”Exciting to read” is a criterium to see that the article is also fun to read for a human. At last, the ”Confabulations” criterium is to be certain that the model does not introduce content that it does not have information on (e.g. write about the wind affecting the race if there is no information about the wind) (Berrios, 1998).

3. Experimental Results

3.1. Results

The raw results of the experiment can be found in Appendix C. When calculating the average for each combination of prompt engineering technique and input format (see Table 3), we can see that Zero-shot prompting with CSV files generated the highest quality articles. One-shot prompting, Few-shot prompting, and CoT performed similarly, while Auto CoT performed significantly worse. In regard to input type, we see CSV and Trainer notes perform similarly, while HTML achieves a lower score.

	CSV	HTML	Notes	Prompting AVG
Zero-shot	8.8	7.0	8.2	8.0
One-shot	7.6	6.4	7.6	7.2
Few-shot	7.8	6.2	7.6	7.2
CoT	7.6	6.8	8.0	7.5
Auto CoT	5.6	7.0	6.4	6.3
Input AVG	7.5	6.7	7.6	–

Table 3. Evaluation Results

When looking at the different criteria seperately (see Table 4), we can see the model is performing well in mentioning all athletes. The factual correctness is also scoring decent, although it does have a higher standard deviation, meaning it varies more between articles. The fluency of the article is average, but we can especially see shortcomings regarding how exciting the articles are to read and the amount of confabulations.

Criterium	Average score	Std
Athletes mentioned	1.97	0.16
Factually correct content	1.68	0.52
Fluent article	1.41	0.59
Exciting to read	1.15	0.69
Confabulations	1.03	0.73

Table 4. Scores per criterium

While evaluating the generated articles, we also made some observations which are not captured by the criteria, but are nonetheless worth mentioning. In general, the output tends to be overly positive on performance, and is very unlikely to be critical of lesser results. Another observation that was made, is the LLM having difficulty when there are lots of results to write about. Here, the LLM often ends up only listing results (sometimes using bullet points), instead of describing them vividly. This was seen most when using HTML or trainer notes with lots of details as input, although it was seen for the other three techniques as well. Furthermore, the LLM tends to confabulate about the weather, regardless of prompt engineering technique. This is also true for the model trying to come up with a slogan for Scopias Atletiek (e.g. the article ending with ”Scopias Atletiek - Affecting your athletic journey”). Lastly, an issue that is seen in some of the articles is regarding generated highlights of a competition. These highlights are not always fair, as some athletes are mentioned in highlights, while others are not, despite similar results.

There were also some observations that were prompt engineering technique specific. One of the most notable ones was the LLM coming up with quotes from trainers or athletes itself when using Auto CoT. This is caused by the chain of thought as generated by the LLM including gathering quotes as one of the steps in generating the article.

Another prompt engineering specific issue which was seen when using One-shot or Few-shot prompting, is results from the provided examples sometimes ending up in the generated articles. It seems like the LLM has difficulty here in separating results from the input from results from the provided examples

Lastly, an issue with trainer notes as input was seen. These notes often include the name of the trainer, which is then included at the end of the article (e.g. Kind regards, [Trainer name]). In a professional article, this should not appear.

4. Discussion

Our study also has some limitations that are worthy to discuss. Presumably, the biggest limitation is the evaluation process. We are aware that evaluating 75 articles by hand is a subjective process that can easily lead to evaluation errors, stemming from humanly bias. However, due to time constraints and the amount of forecasted work, made us decide not to peruse creating an evaluation system with quantifiable criteria that could be used to develop an automated way to evaluate the articles. Furthermore, during the generation of the articles, some of the articles yielded completely unexpected results, not at all with the information from the input. The output here was a mix of random characters. Re-generating of these articles, solved this problem for our study, but likely the model contains some issues. Although not observed, these issues could have influenced the result of the article generation. Another limitation is the prompt injection and the related security aspect. In our research, the prompt injection was under our control (i.e. athletes were not able to prompt the model to write an article). Our focus was therefore less on stress testing the model to find weaknesses and to develop a robust model that can not be tricked easily in providing misleading or inaccurate outputs, but more on stress testing the model on different types of input.

Our study was conducted within a limited time frame, leaving several aspects open for further exploration in future research. Future work could focus on the input, stress testing and evaluation. For the input, research could be conducted on the quality of the input among the same types of input and how this influences the article generation (e.g. does more data lead to better articles?) or how can pre-processing the input with the model lead to better results (e.g. does the quality of the article improve if HTML input is converted to CSV by the model first, subsequently using the CSV input for article generation). Future work could also focus on more comprehensive stress testing using a variety of prompt styles. For real-world applications, further stress testing is essential to prevent the introduction of intentional biases. For instance, when athletes are able to also create prompts, they can request the model to elevate their performance, increase positivity toward themselves or certain athletes, the exclusion of specific athletes, or adjustments of particular scores. The evaluation process is another limitation in our study, and thus also an area where future research could focus on. For example to enhance the evaluation criteria in way that would quantify the criteria and automate the evaluation process.

5. Conclusion

In this project, we researched the ability of LLMs to generate athletics articles. We combined different input formats (CSV, HTML, and Trainer notes) with different prompt engineering techniques (Zero-shot, One-shot, Few-shot, CoT and Auto CoT) to generate articles and evaluated them based on 5 criteria.

Based on this evaluation, we are able to conclude that LLMs are able to produce factual correct articles most of the time. However, we must conclude that the results do vary between articles and we thus cannot rely on LLMs just yet, meaning a human check is still necessary.

The input of the data influences the generated articles, with CSV and Trainer notes outperforming HTML files. This shows that the model has difficulty with the noise which is still included in the HTML, which is not present in CSV files and Trainer notes.

Different prompting techniques have different effects on the output of the LLM. Zero-shot performs best, followed by One-shot, Few-shot and CoT. Auto CoT performs the worst in generating the articles.

In conclusion, a combination of CSV input with Zero-shot prompting generated the best articles for athletics events. Until the reliability of output has improved, however, a human check is still necessary.

References

Arawjo, I., Swoopes, C., Vaithilingam, P., Wattenberg, M., and Glassman, E. L. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing, 2024.

Berrios, G. E. Confabulations: A conceptual history. *Journal of the History of the Neurosciences*, 7(3): 225–241, 1998. doi: 10.1076/jhin.7.3.225.1855. URL https://www.tandfonline.com/doi/abs/10.1076/jhin.7.3.225.1855. PMID: 11623845.

Brown, T. B. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chiang, S.-H., Chao, L.-W., Wang, K.-D., Wang, C.-C., and Peng, W.-C. Badge: Badminton report generation and evaluation with llm. *arXiv preprint arXiv:2406.18116*, 2024.

DAIR.AI. Prompting techniques — prompt engineering guide. https://www.promptingguide.ai/techniques, 2024. Accessed: 2024-11-09.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2. Minneapolis, Minnesota, 2019.

LLama Team, AI @ Meta. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Radford, A. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Sprague, Z., Ye, X., Bostrom, K., Chaudhuri, S., and Durrett, G. Musr: Testing the limits of chain-of-thought with multistep soft reasoning, 2024. URL https://arxiv.org/abs/2310.16049.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., and Fan, Z. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A. Prompts used

A.1. Prompt for generating the Chain of Thought

You are tasked with writing a report about the athletics match. You are given the results of the match and do not have access to additional data. Come up with a chain of thought to write the report. Include which steps should be taken to write the report and what information should be included. Output should be a numbered list of steps.

A.2. Prompt for generating the article

You are a professional copywriter tasked with writing a report about an athletics match, in which Scopias Atletiek participated. The report will be published on the Scopias Atletiek website. Your goal is to write a report that is informative and engaging for the readers. It should thus not simply be a sum up of the results of the match, but also be a pleasure to read. All results of Scopias Atletiek athletes should be included in the report.

Your target audience is the members of Scopias Atletiek, as well as other athletics enthusiasts. The report should be written in a professional and engaging tone, and should be easy to read and understand. Output the report in a markdown format.

If the prompt engineering technique used is Chain of Thought

When writing the report follow this chain of thought:

1. Carefully read the data and extract the results of all athletes from Scopias Atletiek.

2. Write an introduction to the report in which you include what the name of the match was and where it took place.

3. Write a summary of the match, in which you include the results of Scopias Atletiek athletes.

4. Write a conclusion in which you summarize the results of the match and give your opinion on the performance of the athletes.

If the prompt engineering technique used is Auto Chain of Thought

When writing the report follow this chain of thought: `<Generated Chain of Thought>`

If the prompt engineering technique used is One-shot or Few-shot

To help you write the report, here are some examples of previous reports:

Example `X`: `Example X contents`

Finally, all prompts end with the data

Data: `Data`

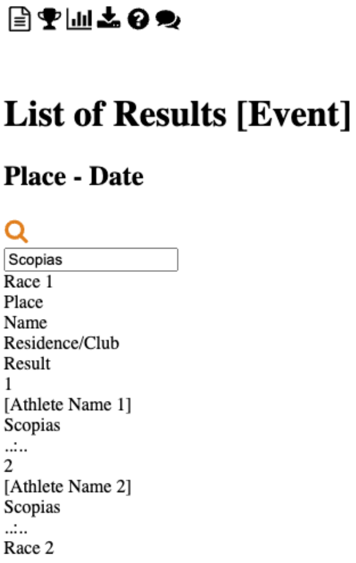
B. Examples of data input

B.1. CSV Input

Race	Place	Athlete	Result
Women U16 (2100m)	1	Athlete 1	39:39
Women U16 (2100m)	2	Athlete 2	40:14
Women U16 (2100m)	3	Athlete 3	40:59
Men U12 (1300m)	1	Athlete 4	12:25
Men U12 (1300m)	2	Athlete 5	12:39
Men U12 (1300m)	3	Athlete 6	13:45

Table 5. Example of CSV input for a random event at a random date

B.2. HTML Input



List of Results [Event]

Place - Date

Scopias
Race 1
Place
Name
Residence/Club
Result
1
[Athlete Name 1]
Scopias
..:..
2
[Athlete Name 2]
Scopias
..:..
Race 2

Figure 2. Example of HTML input from Uitslagen.nl

B.3. Raw Text Input

Notes from a trainer received by email:

”Hi media team!

I wanted to share the highlights from the **[Event]**, where our Scopias athletes performed well across a range of categories.

In the **[Race]**, **[Athlete Name 1]** won with an impressive **xx** minutes and **xx** seconds. Not far behind, **[Athlete name 2]** clocked in at **xx:xx**, securing second place. Meanwhile, in the **[Race]** category, **[Athlete Name 3]** also won with a time of **xx:xx**.

Kind regards,

[Trainer]”

C. Raw results

Input type	Prompt engineering technique	Sample	Athletes mentioned	Factual correct content	Fluent article	Exciting to read	Confabulations	Total score
CSV	Zero-shot	1	2	2	2	2	1	9
CSV	One-shot	1	2	1	2	2	0	7
CSV	Few-shot	1	2	1	2	2	0	7
CSV	Chain of Thought	1	2	2	2	1	1	8
CSV	Auto Chain of Thought	1	2	1	2	1	0	6
CSV	Zero-shot	2	2	1	2	2	2	9
CSV	One-shot	2	2	2	2	1	2	9
CSV	Few-shot	2	2	1	2	2	1	8
CSV	Chain of Thought	2	2	2	1	1	1	7
CSV	Auto Chain of Thought	2	2	1	2	1	0	6
CSV	Zero-shot	3	2	1	2	2	1	8
CSV	One-shot	3	2	2	2	2	1	9
CSV	Few-shot	3	2	2	2	1	2	9
CSV	Chain of Thought	3	2	2	2	1	1	8
CSV	Auto Chain of Thought	3	2	1	1	0	0	4
CSV	Zero-shot	4	2	1	2	2	2	9
CSV	One-shot	4	2	2	2	1	1	8
CSV	Few-shot	4	2	2	2	1	1	8
CSV	Chain of Thought	4	2	2	2	1	1	8
CSV	Auto Chain of Thought	4	2	1	2	1	0	6
CSV	Zero-shot	5	2	2	2	2	1	9
CSV	One-shot	5	1	0	2	2	0	5
CSV	Few-shot	5	2	2	1	0	2	7
CSV	Chain of Thought	5	2	2	1	0	2	7
CSV	Auto Chain of Thought	5	2	2	1	0	1	6

Input type	Prompt engineering technique	Sample	Athletes mentioned	Factual correct content	Fluent article	Exciting to read	Confabulations	Total score
HTML	Zero-shot	1	2	0	1	1	1	5
HTML	One-shot	1	2	2	2	1	0	7
HTML	Few-shot	1	2	1	2	1	1	7
HTML	Chain of Thought	1	2	2	2	1	1	8
HTML	Auto Chain of Thought	1	2	2	2	1	1	8
HTML	Zero-shot	2	2	2	2	1	1	8
HTML	One-shot	2	2	2	1	1	1	7
HTML	Few-shot	2	2	2	1	0	1	6
HTML	Chain of Thought	2	2	1	1	1	1	6
HTML	Auto Chain of Thought	2	2	2	1	0	1	6
HTML	Zero-shot	3	2	2	2	1	2	9
HTML	One-shot	3	2	2	1	0	2	7
HTML	Few-shot	3	2	2	1	1	1	7
HTML	Chain of Thought	3	2	1	1	0	2	6
HTML	Auto Chain of Thought	3	2	2	1	1	2	8
HTML	Zero-shot	4	2	2	1	2	1	8
HTML	One-shot	4	2	2	1	2	0	7
HTML	Few-shot	4	2	2	1	2	1	8
HTML	Chain of Thought	4	2	2	2	1	2	9
HTML	Auto Chain of Thought	4	2	2	1	2	1	8
HTML	Zero-shot	5	2	2	0	0	1	5
HTML	One-shot	5	2	2	0	0	0	4
HTML	Few-shot	5	1	2	0	0	0	3
HTML	Chain of Thought	5	2	2	0	0	1	5
HTML	Auto Chain of Thought	5	2	2	1	0	0	5

Input type	Prompt engineering technique	Sample	Athletes mentioned	Factual correct content	Fluent article	Exciting to read	Confabulations	Total score
Trainer notes	Zero-shot	1	2	2	2	2	2	10
Trainer notes	One-shot	1	2	2	2	2	0	8
Trainer notes	Few-shot	1	2	2	2	2	0	8
Trainer notes	Chain of Thought	1	2	2	2	2	2	10
Trainer notes	Auto Chain of Thought	1	2	2	2	2	0	8
Trainer notes	Zero-shot	2	2	1	1	2	2	8
Trainer notes	One-shot	2	2	1	1	2	2	8
Trainer notes	Few-shot	2	2	1	2	2	2	9
Trainer notes	Chain of Thought	2	2	1	1	2	2	8
Trainer notes	Auto Chain of Thought	2	2	1	1	1	0	5
Trainer notes	Zero-shot	3	2	2	1	1	1	7
Trainer notes	One-shot	3	2	2	1	1	1	7
Trainer notes	Few-shot	3	2	1	1	1	1	6
Trainer notes	Chain of Thought	3	2	2	1	1	1	7
Trainer notes	Auto Chain of Thought	3	2	2	1	1	0	6
Trainer notes	Zero-shot	4	2	2	1	1	2	8
Trainer notes	One-shot	4	2	2	1	1	0	6
Trainer notes	Few-shot	4	2	2	1	2	1	8
Trainer notes	Chain of Thought	4	2	2	1	1	2	8
Trainer notes	Auto Chain of Thought	4	2	2	1	1	2	8
Trainer notes	Zero-shot	5	2	2	2	1	1	8
Trainer notes	One-shot	5	2	2	2	1	2	9
Trainer notes	Few-shot	5	2	2	1	1	1	7
Trainer notes	Chain of Thought	5	2	2	1	1	1	7
Trainer notes	Auto Chain of Thought	5	2	1	1	1	0	5