IE Final Project Report

HAYDEN MCCORMICK

May 2024

1 Project Introduction

Fact-checking articles such as Snopes and PolitiFact currently serve as the de-facto mechanism for verifying or correcting popular claims, especially those made by influential or notorious people. While these articles are excellent resources for those who seek them out, they generally do not have the reach or breadth of automatic systems such as those based on the FEVER dataset [3], which have the ability to dynamically spot potentially dubious claims and fact-check them against external knowledge repositories.

Factoring Fact-Checks[1] is a publication which introduces a middle-ground to bridge these two types of systems; in this paper, the authors developed a BERT-based sequence-to-sequence architecture for detecting the claim, claimant, and verdict (the "factors") of the article. These factors are drawn from the ClaimReview format, a standardized JSON-based schema for annotating fact-checks. This paper, released in 2020, precedes the current wave of powerful large language models, which have shown the potential to beat previous benchmarks in a wide range of NLP tasks. For this project, I experiment with various LLM techniques for tackling the problem, and compare results to those reported in the source paper.

2 Preprocessing

A major area of effort in the project modeling is preprocessing of the ClaimReview dataset used for training.

First, since the data only contains article URLs, the main text of the article needs to be extracted. The authors, most of whom are Google employees, use a private Google tool for accessing pages and scraping the main article text, bringing the total instances of parsable data to 5,868 (694/587/587 test/dev/train split). In my attempt to reproduce this, I ran into two problems: first, the common data source *The Weekly Standard* has since been bought out, and its articles are no longer accessible via their original URL. Second, the article text is non-trivial to find, since different data sources use different HTML structures. After accounting for both of these issues (using the *newspaper* library for the latter task) and filtering out non-English articles, I was able to reconstruct the dataset to 4,497 instances: a 3597/450/450 split.

To ensure the model's predictions could actually be made based on the content of the text, the authors developed a method for correcting factor annotations that do not match exactly with a span of text in the article. To do this, each factor is fuzzy matched with paragraphs of text that contain a certain threshold of overlap. Then, the minimum window substring of overlap is found and specified as the corrected factor. After implementing this, the resulting dataset was ready for evaluation and fine-tuning.

3 Modeling

In my experiments, I hoped to explore the strengths of both local and cloud-based LLMs, and compare the performance of each. Performance was measured using ROUGE score, a metric of unigram overlap popular with summarization tasks, across each individual factor.

3.1 Cloud

Claude-3, which made headlines by claiming so surpass GPT-4 in almost all benchmarks, was the first model I experimented with. For the sake of consistency in evaluation, I limited the test space of zero-shot models to 450 – the same number of samples as the test split of fine-tuned models.

		Claim Rouge-1			Claimant Rouge-1			Verdict Rouge-1		
		F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Cloud	Claude-3 Haiku	**0.84328**	**0.865173**	**0.841905**	0.684582	0.602097	0.924983	0.956650	0.956894	0.959114
	Claude-3 Haiku (modified prompt)	0.772235	0.803173	0.795647	0.588952	0.508332	0.825513	0.887504	0.885195	0.928182
	GPT 3.5	0.672681	0.694470	0.708058	0.700502	0.671399	0.79393	0.742470	0.693237	0.929384
	GPT 3.5 Few-Shot CoT	0.698360	0.792114	0.666542	0.767464	0.759727	0.786857	0.950225	0.950766	0.950650
"Local"	LLaMA un-tuned	0.689685	0.760867	0.673030	0.622603	0.55295	0.810765	0.751739	0.735703	0.886099
	LLaMA fine-tuned	0.75582	0.791943	0.780333	0.849847	0.853452	0.85086	**0.965667**	**0.967004**	**0.970869**
	Mistral fine-tuned	0.71584	0.737471	0.777516	**0.861281**	**0.863988**	**0.862809**	0.95934	0.963074	0.96121
	BERT (Factoring Fact Checks)	0.646	0.664	0.652	0.839	0.852	0.834	0.941	0.944	0.940

Table 1: Modeling results on ClaimReview test set. Best results are highlighted in bold.

The first metric was zero-shot learning using a relatively simple system prompt explaining the task. From this alone, Claude-3 Haiku was able to score the highest Claim metrics of any method, as well as competitive Verdict metrics. However, as we will see, different models are not monotonically better at the task, but instead excel at different aspects: we can see that pattern in this case with the somewhat low Claimant scores. Noting the high recall, I attempted to modify the prompt with the following text:

```
EVERY WORD SHOULD COME EXACTLY FROM THE TEXT! Do not paraphrase. Do not edit
the text in any way.
```

Surprisingly, this noticeably lowered all metrics. Examining the output, the model appears to be selecting significantly longer spans of text, and often chooses ones that do not contain the actual factor it is looking for.

Finally, I experimented with both unmodified GPT-3.5 and a GPT run containing few-shot chain-of-thought prompting. For the latter, rather than simply guessing and iteratively improving prompts, I used DSPy[2], a python library which uses pseudo-Torch building blocks to automatically optimize the prompt of an LLM, given a small set of "training"/bootstrap samples. This significantly improved the performance of GPT; although still lost out on all metrics with the exception of Claimant to Claude-3.

3.2 Local

Using the term "local" LLM loosely (I still accessed these via cloud APIs), I also wanted to compare the performance to that of both base and fine-tuned smaller LLMs.

First, LLaMA-3 7b's performance was expectedly mediocre, although it still beat out GPT's performance in some aspects. However, fine-tuning both LLaMA and Mistral on the held-out training data drastically improved their performance, to the point where each scored the highest aggregate ROUGE-1 scores of a factor type (Verdict and Claimant respectively). For a significantly smaller model, this is extremely promising compared to the massive GPT and Claude-3.

4 Conclusion/Limitations

Implementing LLM-based methods for "factoring fact checks" proved to be extremely impressive, with all BERT metrics being beat out by most newer LLMs. Most surprisingly, even smaller, 7B/8B parameter models can be extremely effective at this task, and each model shows a different skill level for each task; few-shot GPT 3.5, for example, is extremely good at Verdict detection, and mediocre at all other tasks.

Future work could include experimentation with larger amounts of data, as well as tweaking of fine-tuning and prompting methods (DSPy alone has many features and techniques I did not experiment with). In general, a "good" system at this task is one with balanced performance for each class. On this dataset, fine-tuned LLaMA or Claude-3 Haiku likely fit this description the best, though future experimentation may reveal yet more capable models.

References

[1] Shan Jiang, Simon Baumgartner, Abe Ittycheriah, and Cong Yu. Factoring fact-checks: Structured information extraction from fact-checking articles. In *Proceedings of The Web Conference 2020*, WWW '20, page 1592–1603, New York, NY, USA, 2020. Association for Computing Machinery.

[2] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023.

[3] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.