DS-UA 301 Final Project: Report

Michael Kazarian MK8391@NYU.EDU

Marie Qi MQ2066@NYU.EDU

Milestone 3

August 12, 2025

1. Introduction

The main limitation of large language models we seek to address is hallucination, in particular regarding evaluation of misinformative political claims. A promising direction to address this limitation is to augment LLMs with external tools—such as knowledge APIs—that allow them to retrieve and incorporate verified evidence during reasoning. This project uses a logical framework based on scores for corroboration, bias-checking, and evidence strengh to guide llm reasoning and mitigate possible hallucinations by using article retrieval.

2. Methodology

2.0.1 Dataset

We use the LIAR dataset. It contains 12,836 short political claims with six labels (pants-on-fire, false, barely-true, half-true, mostly-true, true) plus metadata.

- Binarization: pants-on-fire, false, barely-true → false; half-true, mostly-true, true → true.
- Test subset: First 300 items from LIAR's designated test split (810 items).
- Preprocessing: Claims are short (1–2 sentences) and often require background context; labels are imbalanced toward true in our 300-item slice, so we report accuracy and F1 and include confusion matrices.

2.0.2 Prompts and Custom metrics

The prompts we use are based on the methods of human fact checkers which we broke down into three main steps where we look for three custom metrics: Credibility score, Bias score, and Evidence Strength.

- Corroboration: Do other sources agree with this statement?
  - We use Serpapi to retrieve 10 articles related to the claim
  - We ask GPT4 to classify a set of articles on the same topic as corroborating, contradicting or neutral.
  - We calculate Credibility score as

$$\frac{N_{\text{corroborating}}}{N_{\text{corroborating}} + N_{\text{contradicting}}}$$

  - Rationale: Mirrors evidence agreement signals popular in human fact-checking

- Bias-checking: Does this source have certain political leanings?
  - Political leaning: GPT-4 assigns Left, Right, Center, Mixed. We map these to a Political Index: Left = 1, Right = +1, Center = 0, Mixed = 0.
  - Tone: GPT-4 assigns Emotional, Neutral. We map to Tone Score: Emotional = 1, Neutral = 0 (binary sentiment proxy).
  - Metric: Bias = Political Index + Tone Score

- Evidence-based Reasoning: Is there strong evidence to support this claim?
  - We ask GPT4 to classify supporting evidence as strong or weak.
  - Evidence Strength is calculated as:

$$\frac{N_{\text{strong}}}{N_{\text{strong}} + N_{\text{weak}}}$$

- Final Trust Score:
  - The trust score is calculated as

$$\text{Trust score} = w_1 * \text{credibility score} + w_2 * \text{1-bias score} + w_3 * \text{evidence score}$$

  - if above .5 we classify as true and weights are decided by the llm
  - Weights: In this milestone, w 1 , w 2 , w 3 w 1 ,w 2 ,w 3 are prompt-set and LLM-instantiated (reported in the JSON output)
  - Predictions are evaluated with sklearn.metrics. Confusion matrices are reported per class.

2.0.3 Baseline system

The baseline model prompts the LLM (GPT-3.5) directly with the claim and asks for a binary truth judgment along with justification. It does not have access to any tools or external sources.

2.0.4 Tool-augmented system

The tool-augmented model uses the ReACT framework to iteratively reason and retrieve information using external tools. This system was implemented using the LangChain library. Specifically, the model has access to a Wikipedia API tool, which returns summaries of information from that site of relevant articles based on search queries. (We also considered a NewsAPI tool, which was tested in earlier versions but excluded from the final procedure due to sparse retrieval and rate limits.)

2.0.5 Tool-augmented system+custom prompting

Another tool augmented system uses Serpapi to scrape for articles relevant to the claim and we use GPT4 instead of 3.5 to assess for corroboration. Based on this we get our credibility, bias, and evidence strength scores.

2.1 Agent configuration

2.1.1 Prompting

Carefully crafted system prompts were used to standardize task framing and minimize prompt bias. Especially important was to word the prompt in such a way that a final prediction was made as often as possible, even when the claim may have been unclear or the evidence inconclusive.

2.1.2 Tool execution

Using the tool, the ReACT agent dynamically formulates queries and integrates retrieved information into its reasoning process before making a final binary prediction. The ReACT agent is technically allotted a maximum of 7 "iterations" (reasoning chains), but our prompt explicitly limited it to 5 (which was generally though not always followed). We set a limit in the first place to escape the problems of indefinitely long reasoning flows (which was generally well managed). To stay under the model's token limits, on the other hand, the agent is limited to a maximum of 3 retrievals per query, and the retrieved evidence (source text from Wikipedia) is no longer than 400 characters.

3. Results

The performance of the baseline and tool-augmented models was evaluated on a test set of 300 claims. Contrary to our initial hypothesis, the baseline model outperformed the tool-augmented system across all major evaluation metrics. The baseline model achieved an accuracy of **47.3%**, with a weighted F1-score of **0.408**, whereas the tool-augmented model reached only **43.3%** accuracy and a weighted F1-score of **0.333**. This represents a 4 percentage point drop in accuracy for the tool-augmented system, which is significant though not extraordinary.

Per-class analysis revealed that both models exhibited strong recall for the True class (0.92), but struggled significantly on the False class, particularly in recall (0.18 for baseline, 0.10 for tool-augmented). While the tool-augmented model had better balance in macro precision, its performance on the False class was markedly weaker in both recall and F1-score, contributing to its lower overall performance. The relevant confusion matrices are on the next page.

The custom prompted model achieved as accuracy of **.69** and f1 score of **.73** suggesting the model is good at detecting positive classes. This result is significantly above the baseline and the tool augmented model. This is an example of an output:
Retrieving articles for claim: Says the Obama administration spent taxpayer dollars on electric cars in Finland (and) windmills in China. Retrieved 10 evidence snippets. Analyzing claim...

CLAIM: Says the Obama administration spent taxpayer dollars on electric cars in Finland (and) windmills in China.
GROUND TRUTH LABEL: barely-true
GPT-4 RESULT:

”credibility score”: 0.2,
”bias score”: {
”political”: ”Mixed”,
”sentiment”: ”Neutral”
},
”evidence strength”: ”Moderate”,
”trust score”: 0.3,
”result”: false,
”summary”: ”The claim that the Obama administration spent taxpayer dollars on electric cars in Finland and windmills in China is mostly false. While there were funds allocated to international projects, the specific claim about electric cars in Finland and windmills in China is not supported by the majority of the evidence.”,
”counts”: {
”n corroborating”: 2,
”n contradicting”: 7,
”n strong evidence”: 3,
”n weak evidence”: 6
}
}

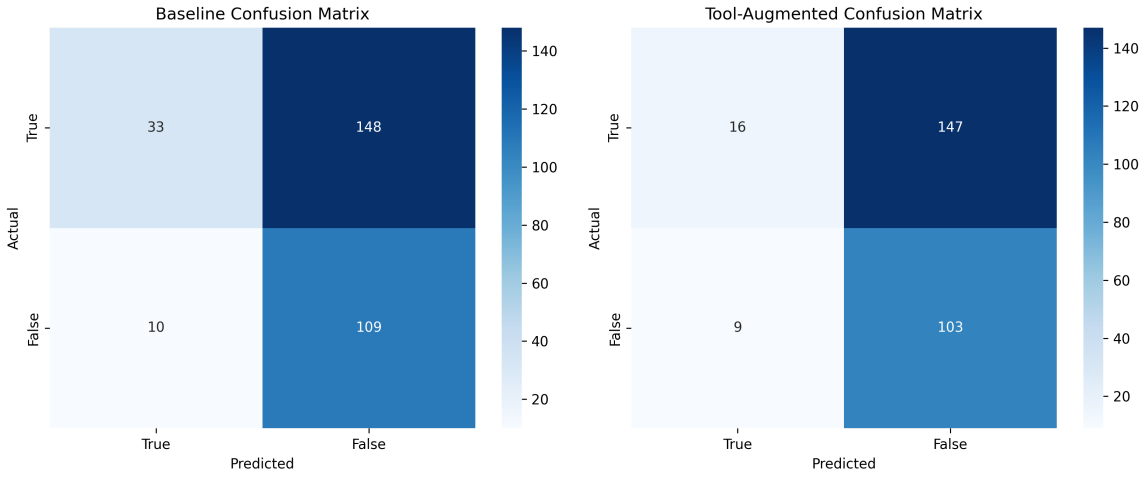

Figure 1: Confusion matrix for binary classification showing TP, FP, FN, and TN.

3.0.1 Qualitative analysis of differential outcomes

Based on our manual, qualitative analysis of the results, specifically the respective justifications given by the baseline and tool-augmented model, the tool-augmented model underperformed in some cases due to its inability to retrieve relevant evidence, which led it to default to incorrect conclusions, mostly when supporting documents were not found. In contrast, the baseline model — despite not consulting external sources — made more accurate predictions in certain historical or well-known cases, due to the richness of the LLM's

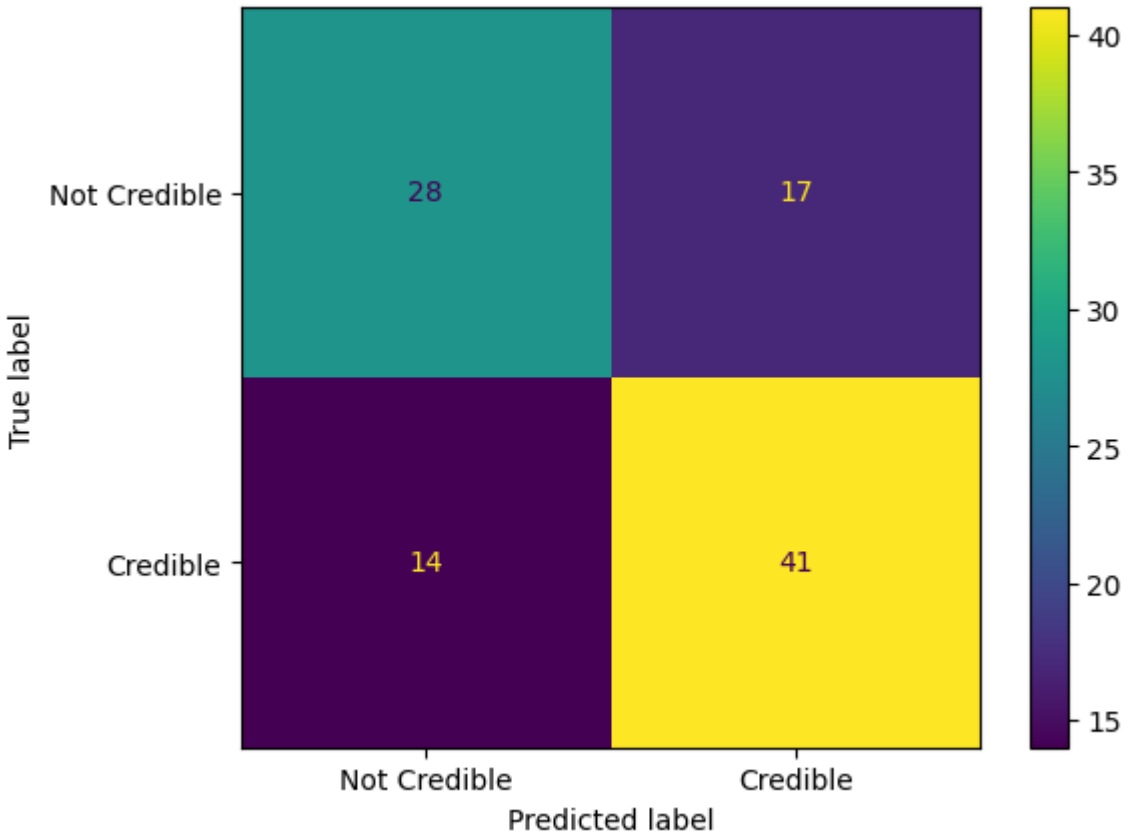

Figure 2: Confusion matrix for binary classification showing TP, FP, FN, and TN.

internal training data. Essentially, the tool-augmented model leaned too heavily on its tool, despite no explicit suggestion in our prompting to do so. Here are a couple of exemplary cases:

> **Claim:** Says President Obama and his National Labor Relations Board sued Boeing over its decision to open a plant in South Carolina.
> **Ground Truth:** True
> **Baseline Prediction:** True
> **Tool Prediction:** False
> **Explanation:** The tool-augmented agent failed to retrieve relevant historical context, while the baseline LLM correctly recalled the 2011 NLRB complaint from training data.

> **Claim:** Says Mitt Romney flip-flopped on an assault weapons ban.
> **Ground Truth:** True
> **Baseline Prediction:** True
> **Tool Prediction:** False
> **Explanation:** The tool agent likely could not find a direct quote or document

and thus marked it false, while the baseline LLM remembered the controversy.

3.1 Custom prompting

The custom prompted model gave classifications based on quantitative analysis so there was theoretically less opportunity to hallucinate false information. We relied mostly on the sentiment analysis and semantic similarity recognition capabilities of GPT, which are considered state of the art, to produce scores for different categories, which may have led to a higher accuracy because it relies less on the model's critical thinking skills and plays to the strengths of llms.

4. Conclusions

4.0.1 Limitations

The dataset itself did not time-stamp claims, so the llm is applying its knowledge base against statements whose truth value may have changed since they were made. -the Wikipedia tool relies on keyword-based queries and can return irrelevant or overly general summaries. In addition, the fact that the tool-augmented system performed worse than the baseline indicates that evidence integration introduces noise or verbosity that degrades final predictions. Our binary True/False labeling oversimplifies many real-world claims, which are context-dependent, ambiguous, or partly true. The Serpapi tool also was limited to 10 articles and increasing this amount could improve accuracy.

4.0.2 Directions for future research

Using prompting to build a more explainable fact checking system can be refined by using different custom metrics and a different threshold, calculated from the ROC curve. We can also use multiple classes rather than true or false like partially true and partially false and increase the number of retrieved articles.

5. Workflow

- Week 1–2: Michael/Marie: Load datasets, and align them with pipeline
- Week 2–3: Michael/Marie: Run and evaluate baseline prompting model
- Week 3–4: Michael: Use custom prompt on gpt3.5
- Week 4–5: Marie: Use wikipedia tool, then SERP Api, to add to baseline model
- Week 5–6: Michael: Compare and contrast performances

Appendix A: Prompts

Baseline Prompt

```
You are a fact-checking expert. Analyze the following claim and determine if it is
     true or false.
Only answer TRUE or FALSE; no other answer than TRUE or FALSE is allowed!
(Even when the claim seems unclear or the evidence seems inconclusive, still only
    answer TRUE or FALSE.)
Claim: "{claim}"
You MUST format your final response as:
Answer: [True/False]
Justification: [Your reasoning]
```

Wikipedia / ReACT Prompt

```
Analyze this claim and determine if it is TRUE or FALSE.
Only answer TRUE or FALSE; no other answer than TRUE or FALSE is allowed!
(Even when the evidence may be "inconclusive," still only answer TRUE or FALSE.)
If you cannot find sufficient evidence after searching, make your best judgment
    based on available information;
you still MUST answer either TRUE or FALSE. Moreover, if you reach the end of the
    fifth reasoning iteration, you MUST
make a decision at that point.

Claim: "{claim}"

Use the evidence_retrieval tool to gather information, then provide your analysis.

You MUST format your final response as:
Answer: [True/False]
Justification: [Your reasoning based on the evidence]

NOTE that the justification is also mandatory.
```

SerpAPI Scoring Prompt

```
You are a fact-checking assistant. For each claim, your job is to:
1. Review the claim.
2. Classify the following evidence snippets as corroborating the claim or
    contradicting the claim.
3. Count the number of corroborating and contradicting sources.
4. Estimate political bias and emotional tone.
5. Assess evidence strength.
6. Calculate a trust score using:
   Trust Score = w1 * Credibility + w2 * (1 - |Bias|) + w3 * Evidence Strength
7. Decide whether the claim is likely true or false.

Output strictly in JSON with:
```

```
{
  "credibility_score": float (0-1),
  "bias_score": {
    "political": "Left/Center/Right/Mixed",
    "sentiment": "Neutral/Emotional"
  },
  "evidence_strength": "None/Weak/Moderate/Strong",
  "trust_score": float (0-1),
  "result": true/false,
  "summary": "Brief explanation",
  "counts": {
    "n_corroborating": int,
    "n_contradicting": int,
    "n_strong_evidence": int,
    "n_weak_evidence": int
  }
}
Only respond with valid JSON.
```