Enhanced Claim Verification System with Real Source Integration: Algorithmic Design and Mathematical Framework

Technical Documentation

May 23, 2025

Abstract

This report presents a comprehensive technical analysis of an Enhanced Claim Verification System that integrates real-time source gathering with mathematical confidence scoring. The system employs multi-API source aggregation, domain credibility assessment, text analysis algorithms, and a sophisticated confidence calculation framework. Key innovations include weighted credibility scoring, hedging density analysis, source agreement quantification, and adaptive confidence boundaries.

Contents

1 Introduction

The Enhanced Claim Verification System represents a significant advancement in automated fact-checking technology. Unlike traditional systems that rely solely on pre-existing knowledge bases, this implementation dynamically gathers evidence from multiple real-time sources and applies rigorous mathematical models to assess claim veracity and confidence levels.

1.1 System Architecture Overview

The system comprises four primary components:

1. **Real Source Gatherer**: Multi-API evidence collection
2. **Confidence Calculator**: Mathematical reliability assessment
3. **Claim Verifier**: LLM-based verification with structured reasoning
4. **Evidence Analysis**: Automated text analysis and source evaluation

2 Source Evidence Data Structure

2.1 Evidence Representation

Each piece of evidence is represented using a structured dataclass:

$$E = \{T, C, S, \text{Content}, U, \text{Title}, A, D, N\} \tag{1}$$

where:

$$T = \text{source\_type} \in \{\text{academic, web, news, expert}\} \tag{2}$$
$$C = \text{credibility\_score} \in [0, 1] \tag{3}$$
$$S = \text{supports\_claim} \in \{\text{True, False, None}\} \tag{4}$$
$$U = \text{url (optional)} \tag{5}$$
$$A = \text{author (optional)} \tag{6}$$
$$D = \text{publication\_date (optional)} \tag{7}$$
$$N = \text{citation\_count (optional)} \tag{8}$$

3 Multi-Source Evidence Gathering

3.1 Academic Source Integration

The system integrates with Semantic Scholar API for academic evidence:

Algorithm 1 Academic Source Gathering

Require: query string q, limit ℓ
Ensure: list of academic sources S_a
Initialize $S_a \leftarrow \emptyset$
params $\leftarrow \{query : q, limit : \ell, fields : F\}$
response $\leftarrow$ API_CALL(semantic_scholar, params)
for each paper p in response.data **do**
    support $\leftarrow$ ANALYZE_TEXT_SUPPORT(p.abstract, q)
    $C_{base} \leftarrow 0.85$ {High base credibility for academic}
    if p.citationCount > 100 **then**
        $C_{base} \leftarrow \min(0.95, C_{base} + 0.05)$
    else if p.citationCount > 50 **then**
        $C_{base} \leftarrow \min(0.90, C_{base} + 0.03)$
    end if
    Create evidence e with calculated parameters
    $S_a \leftarrow S_a \cup \{e\}$
end for
return S_a

3.2 Domain Credibility Assessment

The system employs a tiered credibility scoring mechanism:

$$C_{domain}(d) = \begin{cases} [0.9, 1.0] & \text{if } d \in T_1 \text{ (Nature, Science, NEJM)} \\ [0.75, 0.89] & \text{if } d \in T_2 \text{ (Reuters, BBC, NPR)} \\ [0.55, 0.74] & \text{if } d \in T_3 \text{ (Wikipedia, CNN)} \\ [0.35, 0.54] & \text{if } d \in T_4 \text{ (Medium, Blogs)} \\ 0.75 & \text{if } d \text{ ends with .gov or .edu} \\ 0.50 & \text{otherwise (unknown domains)} \end{cases} \tag{9}$$

3.3 Web Source Integration

Web sources are gathered via Serper API with domain-based credibility assessment:

Algorithm 2 Web Source Gathering with Credibility Assessment

Require: query q, limit ℓ
Ensure: credibility-weighted source list S_w
$S_w \leftarrow \emptyset$
response $\leftarrow$ SERPER_SEARCH(q, ℓ)
for each result r in response.organic **do**
    $C_{cred} \leftarrow$ ASSESS_DOMAIN_CREDIBILITY(r.url)
    support $\leftarrow$ ANALYZE_TEXT_SUPPORT(r.snippet, q)
    $T_{source} \leftarrow$ CLASSIFY_SOURCE_TYPE(r.url)
    Create evidence e with parameters
    $S_w \leftarrow S_w \cup \{e\}$
end for
return S_w

4 Text Analysis and Support Determination

4.1 Claim Support Analysis

The system analyzes text to determine support for claims using linguistic indicators:

Algorithm 3 Text Support Analysis

Require: text t, claim c
Ensure: support indicator $s \in \{\text{True, False, None}\}$

$t_{lower} \leftarrow \text{LOWERCASE}(t)$
$c_{lower} \leftarrow \text{LOWERCASE}(c)$
$W_c \leftarrow \text{EXTRACT\_KEYWORDS}(c_{lower}) \setminus \text{STOP\_WORDS}$
$r_{score} \leftarrow \frac{|\{w \in W_c : w \in t_{lower}\}|}{|W_c|}$
if $r_{score} < 0.3$ **then**
  return None {Insufficient relevance}
end if
$I_{support} \leftarrow$ count of support indicators in t_{lower}
$I_{oppose} \leftarrow$ count of opposition indicators in t_{lower}
$I_{uncertain} \leftarrow$ count of uncertainty indicators in t_{lower}
if $I_{support} > I_{oppose}$ AND $I_{support} > I_{uncertain}$ **then**
  return True
else if $I_{oppose} > I_{support}$ AND $I_{oppose} > I_{uncertain}$ **then**
  return False
else
  return None
end if

4.2 Linguistic Indicator Sets

The system employs three categories of linguistic indicators:

Support Indicators (I_s): {confirm, support, evidence shows, research indicates, proven, demonstrated, validates}

Opposition Indicators (I_o): {disprove, refute, contradict, false, debunked, no evidence, lacks evidence}

Uncertainty Indicators (I_u): {unclear, mixed evidence, conflicting, debate, controversial, inconclusive}

5 Enhanced Confidence Calculation Framework

5.1 Mathematical Model

The confidence calculation employs a weighted multi-component model:

$$\text{Confidence} = f(\text{Evidence\_Score}, \text{Tone\_Score}) \tag{10}$$

where Evidence Score is computed as:

$$E_{score} = w_n \cdot N_{mapped} + w_c \cdot C_{weighted} + w_a \cdot A_{score} \tag{11}$$

with default weights: $w_n = 0.25$, $w_c = 0.45$, $w_a = 0.30$

5.2 Source Volume Mapping

Source count is mapped using square root scaling for better distribution:

$$N_{mapped}(n) = \begin{cases} 0 & \text{if } n \leq 0 \\ 1 & \text{if } n \geq n_{max} \\ \frac{\sqrt{n}}{\sqrt{n_{max}}} & \text{otherwise} \end{cases} \tag{12}$$

where $n_{max} = 8$ is the maximum effective source count.

5.3 Weighted Credibility Calculation

Credibility is calculated with higher weights for decisive sources:

$$C_{weighted} = \frac{\sum_{i=1}^{|S|} C_i \cdot W_i}{\sum_{i=1}^{|S|} W_i} \tag{13}$$

where:

$$W_i = \begin{cases} 1.2 & \text{if source } i \text{ is decisive (supports/opposes)} \\ 1.0 & \text{if source } i \text{ is neutral} \end{cases} \tag{14}$$

5.4 Agreement Score Calculation

The agreement score quantifies source consensus:

5.5 Hedging Density Analysis

Hedging words indicate uncertainty and reduce confidence:

$$H_{density} = \min\left(\frac{|W_h \cap W_t|}{|W_t|}, 0.15\right) \tag{15}$$

where W_h is the set of hedging words and W_t is the set of all words in the text.

Hedging Words Set (W_h): {may, might, could, possibly, likely, suggest, appear, indicate, perhaps, probably, potentially, seems, unclear, uncertain, ambiguous, debatable}

The tone score is calculated as:

$$T_{score} = 1.0 - H_{density} \tag{16}$$

5.6 Final Confidence Computation

The final confidence employs adaptive weighting based on evidence quality:

$$\text{Final\_Score} = \begin{cases} 0.8 \cdot E_{score} + 0.2 \cdot T_{score} & \text{if } n \geq 3 \text{ and } C_{weighted} > 0.7 \\ 0.7 \cdot E_{score} + 0.3 \cdot T_{score} & \text{if } n \geq 1 \text{ and } C_{weighted} > 0.5 \\ 0.5 \cdot E_{score} + 0.5 \cdot T_{score} & \text{otherwise} \end{cases} \tag{17}$$

Algorithm 4 Agreement Score Calculation

Require: source list S
Ensure: agreement score $A \in [0, 1]$

```
n_support ← |{s ∈ S : s.supports_claim = True}|
n_oppose ← |{s ∈ S : s.supports_claim = False}|
n_neutral ← |{s ∈ S : s.supports_claim = None}|
n_total ← |S|
n_decisive ← n_support + n_oppose
if n_decisive = 0 then
   if n_neutral > 3 then
      return 0.4 {Many neutral sources suggest complexity}
   else
      return 0.5
   end if
else if n_support ≥ n_oppose then
   A ← n_support / n_total
   if n_support > 0 AND n_oppose = 0 AND n_support ≥ 2 then
      A ← min(1.0, A + 0.1) {Consensus bonus}
   end if
   return A
else
   A ← 0.3 − (n_oppose / n_total) · 0.2
   return max(0.1, A) {Floor at 0.1}
end if
```

5.7 Source Quality Bonus

High-quality sources receive additional weighting:

$$E_{score}^{final} = \begin{cases} \min(1.0, E_{score} + 0.05) & \text{if } |\{s \in S : C_s > 0.8\}| \geq 2 \\ E_{score} & \text{otherwise} \end{cases} \tag{18}$$

5.8 Confidence Boundaries

The final confidence percentage is bounded and interpreted:

$$\text{Confidence\_Percentage} = \max(15, \min(95, \lfloor \text{Final\_Score} \times 100 \rfloor)) \tag{19}$$

Confidence Range	Interpretation
0-20%	Very Low - Highly unreliable
21-40%	Low - Questionable reliability
41-60%	Moderate - Ambiguous evidence
61-80%	High - Generally reliable
81-100%	Very High - Highly reliable

Table 1: Confidence Interpretation Boundaries

6 Claim Classification System

The system classifies claims into categories for optimized processing:

Algorithm 5 Claim Classification

Require: claim string c
Ensure: claim type T_c

```
c_lower ← LOWERCASE(c)
if c_lower contains {study, research, scientist, percent, data shows} then
    return "Scientific/Statistical"
else if c_lower contains {said, stated, according to, quote, claimed} then
    return "Quote/Statement"
else if c_lower contains {happened, occurred, when, date, year, during} then
    return "Historical/Factual"
else if c_lower contains {health, medical, disease, treatment, cure} then
    return "Health/Medical"
else if c_lower contains {will, going to, predict, future, forecast} then
    return "Predictive/Future"
else
    return "General Factual"
end if
```

7 LLM Integration and Structured Reasoning

7.1 Verification Prompt Template

The system uses a structured prompt template for consistent LLM analysis:

```
You are an expert fact-checker. Analyze the following claim and evidence.

CLAIM: {claim}
EVIDENCE FROM SOURCES: {evidence}

Your task:
1. Determine if the claim is TRUE, FALSE, or CAN'T SAY
2. Provide detailed reasoning
3. Consider source credibility and consistency

Guidelines:
- Use TRUE only if evidence strongly supports
- Use FALSE only if evidence clearly contradicts
- Use CAN'T SAY if evidence is mixed/insufficient
- Avoid excessive hedging unless genuinely uncertain

Response format:
VERDICT: [TRUE/FALSE/CAN'T SAY]
REASONING: [Detailed analysis]
```

7.2 Verdict Extraction Algorithm

Algorithm 6 Verdict Extraction from LLM Response

Require: LLM response text R
Ensure: verdict $V \in \{\text{TRUE, FALSE, CAN'T SAY}\}$

```
V ← "CAN'T SAY" {Default verdict}
if "VERDICT:" in R then
    Extract line containing "VERDICT:"
    V_text ← text after "VERDICT:" (uppercase)
    if "TRUE" in V_text AND "FALSE" not in V_text then
        V ← "TRUE"
    else if "FALSE" in V_text then
        V ← "FALSE"
    end if
end if
return V
```

8 Error Handling and Robustness

8.1 API Failure Management

The system implements graceful degradation when APIs fail:

Algorithm 7 Robust Source Gathering

Require: claim c
Ensure: source list S (possibly empty)
$S \leftarrow \emptyset$
$S_{academic} \leftarrow$ SEARCH_SEMANTIC_SCHOLAR(c)
$S \leftarrow S \cup S_{academic}$ Exception e
LOG("Academic search failed: " + e)
$S_{web} \leftarrow$ SEARCH_WEB_SERPER(c)
$S \leftarrow S \cup S_{web}$ Exception e
LOG("Web search failed: " + e)
$S_{expert} \leftarrow$ SEARCH_PERPLEXITY(c)
$S \leftarrow S \cup S_{expert}$ Exception e
LOG("Expert analysis failed: " + e)
return S

8.2 Deduplication Logic

Sources are deduplicated based on URL similarity:

Algorithm 8 Source Deduplication

Require: source list S
Ensure: unique source list S_{unique}
seen_urls $\leftarrow \emptyset$
$S_{unique} \leftarrow \emptyset$
for each source s in S **do**
  url_key $\leftarrow$ s.url or (s.title + s.content[: 50])
  if url_key $\notin$ seen_urls **then**
    seen_urls $\leftarrow$ seen_urls $\cup$ {url_key}
    $S_{unique} \leftarrow S_{unique} \cup \{s\}$
  end if
end for
return S_{unique}

9 Performance Considerations

9.1 API Rate Limiting

The system implements delays between API calls to respect rate limits:

$$\text{Delay} = \max(1.0, \frac{\text{API\_calls\_per\_minute}}{60}) \tag{20}$$

9.2 Concurrent Processing

For batch verification, the system supports concurrent processing with controlled threading:

Algorithm 9 Batch Claim Verification

Require: claim list $C = \{c_1, c_2, ..., c_n\}$
Ensure: result list $R = \{r_1, r_2, ..., r_n\}$
$R \leftarrow \emptyset$
Initialize thread pool with max workers = 3
for $i = 1$ to n **do**
  $r_i \leftarrow$ VERIFY_CLAIM(c_i)
  Sleep(1.0) {Rate limiting} Exception e
  $r_i \leftarrow$ CREATE_ERROR_RESULT(c_i, e)
  $R \leftarrow R \cup \{r_i\}$
end for
return R

10 System Integration Architecture

10.1 Component Interaction Flow

```
Input Claim → Claim Classification → Evidence Gathering
     ↓                                       ↓
LLM Analysis ← Evidence Formatting ← Source Assessment
     ↓                                       ↓
Verdict Extraction → Confidence Calculation → Final Result
```

Figure 1: System Component Interaction Flow

10.2 Data Flow Pipeline

The complete verification pipeline processes claims through multiple stages:

1. **Input Processing**: Claim normalization and classification
2. **Source Gathering**: Multi-API evidence collection with error handling
3. **Credibility Assessment**: Domain-based and content-based scoring
4. **Evidence Analysis**: Text analysis for claim support determination
5. **LLM Verification**: Structured reasoning with evidence synthesis
6. **Confidence Calculation**: Mathematical reliability assessment
7. **Result Compilation**: Comprehensive report generation

11 Experimental Validation

11.1 Test Case Categories

The system includes predefined test claims across multiple categories:

Category	Example Claims
Scientific/Health	"Coffee reduces the risk of type 2 diabetes"
Historical/Factual	"The Great Wall of China is visible from space"
Medical/Safety	"Vaccines cause autism in children"
Environmental	"Climate change is primarily caused by human activities"
Quotes/Attribution	"Einstein said 'Imagination is more important than knowledge'"
Health/Nutrition	"Drinking 8 glasses of water per day is necessary"
Cognitive Science	"The human brain only uses 10% of its capacity"
Animal Behavior	"Goldfish have a 3-second memory span"

Table 2: Test Claim Categories

12 Limitations and Future Enhancements

12.1 Current Limitations

1. **Language Dependency**: Currently optimized for English text analysis
2. **API Dependency**: System reliability depends on external API availability
3. **Temporal Constraints**: Knowledge cutoff limitations for recent events
4. **Domain Specificity**: Credibility assessments may not cover all specialized domains

12.2 Proposed Enhancements

1. **Multi-language Support**: Extend linguistic analysis to multiple languages
2. **Caching Mechanisms**: Implement intelligent caching for frequently verified claims
3. **Real-time Updates**: Integration with news feeds for temporal claim verification
4. **Domain Expansion**: Extended credibility databases for specialized fields
5. **Machine Learning Integration**: Training custom models for claim-specific analysis

13 Conclusion

The Enhanced Claim Verification System represents a sophisticated approach to automated fact-checking that combines real-time source gathering, mathematical confidence modeling, and structured AI reasoning. The system's multi-layered architecture ensures robustness while maintaining high accuracy through rigorous evidence evaluation and confidence quantification.

Key technical contributions include:

- Novel weighted confidence calculation framework
- Comprehensive domain credibility assessment
- Advanced hedging density analysis
- Real-time multi-source evidence integration
- Structured LLM reasoning with error handling

The mathematical frameworks presented provide a solid foundation for reliable claim verification while maintaining transparency in the decision-making process. The system's modular design facilitates future enhancements and adaptations to emerging verification challenges.

14 Appendix: Implementation Details

14.1 API Configuration Requirements

The system requires the following API keys for full functionality:

- **OPENAI_API_KEY** or **GEMINI_API_KEY**: LLM processing
- **SEMANTIC_SCHOLAR_API_KEY**: Academic source gathering
- **SERPER_API_KEY**: Web search functionality
- **PERPLEXITY_API_KEY**: Expert analysis integration
- **JINA_API_KEY**: Optional content extraction

14.2 Configuration Parameters

Parameter	**Default Value**	**Description**
n_{max}	8	Maximum effective source count
w_n	0.25	Source volume weight
w_c	0.45	Credibility weight
w_a	0.30	Agreement weight
Hedging Cap	0.15	Maximum hedging penalty
Confidence Floor	15%	Minimum confidence score
Confidence Ceiling	95%	Maximum confidence score

Table 3: System Configuration Parameters