The Linguist's Lie Detector: Linguistic Knowledge in Large Language Models

Lucía Catalán Gris[1], **Kim Gerdes**[1], **John S. Y. Lee**[2]
[1] Université Paris-Saclay, Lisn, CNRS, Orsay, France
[2] City University of Hong Kong, Department of Linguistics and Translation, Hong Kong SAR, China
lucia.catalan-gris@lisn.fr, kim.gerdes@lisn.fr, jsylee@cityu.edu.hk

Abstract

We present a benchmark and evaluation pipeline for assessing how well large language models (LLMs) handle linguistic knowledge. Starting from a curated subcorpus of 11 syntax-focused articles published in *Glossa: A Journal of General Linguistics* (2016–2026), we design a pipeline that (1) segments article text into sentences, (2) extracts atomic, verifiable statements, and (3) classifies them into linguistic categories (language-specific, typological, theoretical, citation, or structural). Each stage is evaluated against human gold annotations produced by three annotators, with inter-annotator agreement measured via Krippendorff's α and Cohen's κ. We compare several LLMs on extraction and classification, using BERTScore-style similarity for extraction and macro F1 for classification. Finally, we generate contradictions of the true linguistic statements and test whether LLMs can distinguish true from false claims. On a challenge set of 705 linguistic statements, we compare eight LLMs, with Gemini 3 Flash achieving the highest F1 score of 0.66, indicating that current models possess limited but non-trivial linguistic knowledge.

Keywords: fact-checking, linguistics, LLM evaluation, statement extraction, Glossa

1. Introduction

Large language models (LLMs) are trained on vast amounts of linguistic data and perform well on many language tasks. However, whether this success reflects genuine linguistic knowledge remains an open question. Linguistics provides a useful test domain for exploring this question: LLMs are *trained on* language, but are they knowledgeable *about* language? Can they distinguish a true claim about the Warlpiri sentence structure from a plausible but false one?

To illustrate the challenge, consider the following example from our corpus, which includes one true statement alongside two LLM-generated contradictions:

(1) a. In Warlpiri, all intransitive verbs combine with an absolutive subject. [true]
b. In Warlpiri, some intransitive verbs combine with an ergative subject.[1] [false; GPT-5.2]
c. In Warlpiri, some intransitive verbs combine with a non-absolutive subject. [false; Gemini 2.5 Flash]

Both false variants avoid explicit verbal negation yet reverse the truth conditions, creating minimal contrasts that are difficult to detect without relevant linguistic knowledge—exactly the kind of challenge a linguistically competent system should be able to resolve.

By evaluating models on such metalinguistic statements, we test their ability to assess claims about language itself rather than simply generate well-formed text. This provides a controlled way to examine whether LLMs truly understand linguistic rules, rather than just mimicking patterns they've seen before.

In this paper, we present a pipeline for checking LLMs' linguistic knowledge. Our contributions are:[2]

1. A curated dataset of 342 statements from open-access syntax articles from *Glossa: A Journal of General Linguistics*, spanning both high- and low-resource languages.

2. A three-stage NLP pipeline—Sentence Segmentation, Statement Extraction, Statement Classification—with detailed prompts and multi-model comparisons.

3. Gold annotations by three human annotators for both extraction and classification, with measured inter-annotator agreement.

4. A linguistic knowledge evaluation in which LLMs determine whether 705 linguistic statements are true or false; one-third are correct, and two-thirds are deliberately falsified.

[1] While technically false for Warlpiri, this statement is typologically plausible. In many split-intransitive (active-stative) languages, agentive intransitive subjects (*unergatives*) receive ergative marking. Furthermore, Warlpiri "middle" verbs take an ergative subject and a dative object; to an LLM, these might be misclassified as intransitives due to the lack of an absolutive/accusative argument.

[2] All data, prompts, and code are publicly available under CC-BY 4.0 at https://github.com/linguistic-fact-checking/nslp26

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 235–246
12 May 2026.

2. Related Work

This work sits at the intersection of several research areas: automated fact-checking, scientific claim verification, information extraction from scholarly articles, and automatic negation generation. In this section, we briefly review these research areas and discuss how our pipeline complements existing approaches.

2.1. General-Domain Fact-Checking

Automated fact-checking has received considerable attention in the NLP community. Thorne and Vlachos (2018) provided an early survey unifying task formulations across domains—focusing on claim inputs (such as triples or text), evidence retrieval (from structured and unstructured sources), and veracity outputs—noting that scientific journal text and encyclopedia articles are among the most common evidence sources. Guo et al. (2022) extended this survey with a three-stage NLP framework (claim detection, evidence retrieval, claim verification) and identified academic papers as a frequently used textual evidence source. More recently, Chen et al. (2024) developed a realistic pipeline for complex claim verification that retrieves raw web evidence and decomposes claims into sub-tasks: question generation, retrieval, answer extraction, and verdict synthesis.

A key dataset in this task is AVeriTeC (Schlichtkrull et al., 2023), which contains 4,568 real-world claims annotated with question/answer pairs, web evidence, and verdicts (Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking). Putta et al. (2025) introduced ClaimCheck, a fact-checking system evaluated on AVeriTeC that integrates claim-matching with novel claim processing using smaller open-source LLMs, demonstrating 62.6% verdict prediction accuracy.

2.2. Scientific Claim Verification

While general fact-checking typically accepts one or more sentences from a web page (e.g., Wikipedia) as sufficient evidence, scientific fact-checking imposes more rigorous evidentiary standards. Vladika and Matthes (2023) surveyed scientific fact-checking resources and approaches. They highlighted the need for automated tools that can handle the complex language found in research publications and assist scientists in verifying hypotheses and discovering evidence. Pradeep et al. (2021) proposed VerT5erini, a T5-based system for scientific claim verification in the biomedical domain, evaluating on SciFACT and demonstrating generalization to COVID-19 claims using the CORD-19 corpus. Sarrouti et al. (2021) introduced HEALTHVER, a dataset for evidence-based fact-checking of health-related claims against scientific articles, with three verdict types (Supports, Refutes, Neutral).

To our knowledge, **no prior work targets linguistic claims** extracted from peer-reviewed linguistics articles.

2.3. Claim Extraction from Scientific Text

Our Statement Extraction pipeline is motivated by research on identifying and formalizing claims in scientific publications. Early work by Nasar et al. (2018) surveyed computational approaches for automatically extracting structured information from scientific literature, including rule-based methods, conditional random fields (CRFs), and deep learning techniques. Building on this paradigm, Bekoulis et al. (2021) examined the Fact Extraction and VERification (FEVER) framework, including the SciFACT dataset, which validates scientific claims against a corpus of abstracts through abstract retrieval, rationale selection, and label prediction.

Several efforts have targeted the extraction of atomic claims specifically. Blake (2010) introduced the Claim Framework for identifying explicit, implicit, and under-specified claims in biomedical articles, differentiating levels of evidence. Jansen and Kuhn (2016) presented an approach for extracting core claims and representing them as AIDA sentences—atomic, independent, declarative, absolute English statements—providing a normalized format for scientific findings. Bleuze (2024) reported on defining claim types and training models for automated claim detection in NLP research papers, creating an annotated corpus of claim categories. At a larger scale, Dessì et al. (2022) introduced CS-KG, a knowledge graph of 41 million statements automatically extracted from 6.7 million computer science articles.

More recently, LLMs have been applied to structured extraction. Dagdelen et al. (2024) demonstrated that fine-tuned LLMs (GPT-3, Llama-2) can extract complex structured knowledge from scientific text and return it in JSON format, a paradigm closely related to our prompt-based extraction pipeline. Our work differs from these approaches in that we extract *linguistic* claims from full articles (not abstracts), enforce self-containedness and atomicity through detailed prompting, and evaluate against human gold annotations with measured inter-annotator agreement.

A related line of research uses NLP to predict whether scientific claims are replicable (Youyou et al., 2023); our work addresses the complementary question of whether claims are *factually correct* with respect to linguistic data.

2.4. Automatic Negation and Contradiction Generation

A distinctive feature of our evaluation is the use of LLM-generated contradictions to create false linguistic statements.

Bilu et al. (2015) addressed automatic claim negation in the context of argumentation mining, proposing rule-based and statistical methods for generating negations of natural-language claims. While Bilu et al. (2015) demonstrated that rule-based methods involving explicit negation markers (e.g., 'not', 'no') can successfully negate 75% of claims, they identified 'Usability'—the plausibility of a negation in context—as the primary bottleneck. Our approach extends this line of work by using LLMs with a carefully designed prompt that enforces strict logical contradiction while *prohibiting* explicit negation markers ("no", "not"), making the resulting false statements harder to detect through surface cues alone. Early experiments confirmed that this approach produces higher-quality contradictions than rule-based methods, consistent with the general advantage of LLMs on generation tasks.

3. Tasks Definition

3.1. Task 1: Statement Extraction

Given a paragraph, the goal is to extract all the atomic, self-contained, verifiable statements within the text, adhering to these guidelines:

- Each statement must be **atomic**, meaning that coordinated elements are separated whenever each conjunct expresses a distinct, independently meaningful proposition.
- Each statement must be **self-contained**, requiring the resolution of all pronouns and the expansion of abbreviations to ensure that they can be understood without reference to the surrounding context.
- Each statement must **preserve technical precision**. They should retain the technical terminology and explicitly include the relevant language studied when contextually clear. Moreover, linguistic examples should be incorporated directly into the statement text, with example numbers, glosses, and translations removed.
- Pure conditionals should be retained as single statements, whereas causal or inferential chains should yield both the premise and the resulting implication.
- Citation attribution should be preserved, so that if an author endorses a cited claim, both the attributed version and the general version are extracted.

3.2. Task 2: Statement Classification

Given a statement, the task is to assign one of the following labels:

- **Language-Specific statements (L-Spec)**: empirical claims about a specific language that are verifiable on language data.
- **Language-Typological statements (L-Typo)**: cross-linguistic or comparative claims.
- **Language-Theoretical statements (L-Theo)**: theory-internal generalizations that are not directly verifiable on raw data.
- **Citations (C)**: statements attributed to another author or paper.
- **Structural (S)**: structural or methodological statements that do not contain any linguistic claim, such as section headers, definitions, methodology descriptions, meta-discourse.

3.3. Task 3: Linguistic Knowledge Evaluation

The final task analyzes whether LLMs possess genuine linguistic knowledge beyond surface-level pattern matching.

The input consists of a claim previously classified as a linguistic statement (L-Spec, L-Typo, and L-Theo). For each statement, the model is expected to provide:

- A **verdict** (True or False): an assessment of the statement's factual accuracy.
- A **justification**: a concise explanation supporting the verdict.
- A **confidence score**: a numerical estimate of the model's certainty in its verdict. In Section **??** we show that these scores are positively associated with correctness, suggesting that the model is reasonably well calibrated on this task. Therefore, they are useful for selective prediction in a semi-automated workflow: high-confidence judgments can be prioritized for automatic acceptance, whereas low-confidence or uncertain cases can be flagged for human review.

The distinction between L-Spec, L-Typo, and L-Theo, while not strictly necessary for the LLM-based verification methods evaluated in this paper, will facilitate future work in selecting appropriate

linguistic resources for claim verification. For example, L-Spec claims may be verified with UD treebanks for the relevant language, and L-Typo claims can be checked in typological databases such as WALS, Grambank, or APiCS.

4. Dataset Construction

To enable a reliable evaluation of LLM performance, we created a manually curated gold dataset. This section outlines the methodology for producing the gold standard, including corpus collection and sampling, data preparation, and the human annotation process.

4.1. Article Collection and Sampling

We collected 1,139 scientific articles published in *Glossa: A Journal of General Linguistics* between its creation in 2016 and February 2026. Among the most prestigious journals for general linguistics, *Glossa* publishes under a Creative Commons Attribution 4.0 license, making it ideal for open-science research.

From these, we selected the articles that meet three criteria:

- The article has an **XML (JATS) file** available. Out of the 1,139 articles collected, 1,025 fulfilled this constraint.
- The article belongs to the domain of **syntax**, as determined by its title and keywords (693 of the 1,025 articles). Our focus on syntax is motivated by our planned future work on claim verification on real linguistic data, such as syntactic treebanks, to supplement the LLM-based methods presented in this paper.
- The language(s) studied in the article should span the spectrum of high- to low-resource languages, and they should also be represented in **Universal Dependencies** (UD) treebanks, enabling future cross-validation of linguistic claims on treebank data.

This selection yielded **11 articles** written in English, that study both high-resource and low/mid-resource languages.

4.2. Automatic Data Preparation

To prepare for human annotation, we built a small subset of statements using the pipeline presented in Figure 1.

First, we parsed 11 JATS-XML files–the previously sampled articles in the previous section–into sections and paragraphs, resulting in 1,082 unique paragraphs. The linguistic examples cited in the text were included within their enclosing paragraph rather than treated as separate units.

After, we selected a subset of 63 paragraphs, aiming to obtain at least 10 statements per article. Each paragraph was segmented into sentences using an LLM (GPT-5.2). The abstract was treated as a paragraph, and we added the preceding paragraph as context to resolve cross-references. From these 63 paragraphs, we ended up with 108 unique sentences. Each paragraph contained between 6 and 32 sentences, with an average of 12 sentences per paragraph.

For each of the previous extracted sentences, an LLM (Gemini 2.5 Flash) was prompted to do a first approximation to Statement Extraction (Section 3.1) and extract all the atomic, self-contained, verifiable statements. A total of 342 atomic statements were extracted from the 108 unique sentences, averaging 3.17 statements per sentence.

Then, a first approximation of Statement Classification (Section 3.2) was run with Gemini 2.5 Flash. The 342 atomic statements were classified into the categories L-Spec, L-Typo, L-Theo, C, or S. For statements labelled as L-Spec, we tasked the model to identify the language referred to in each statement. For example, the statement "in Warlpiri, all intransitive verbs combine with an absolutive subject" was classified as L-Spec, and the language identified was Warlpiri.

All data preparation stages used the provider default temperature setting (1.0) and the prompts provided in the Appendix A. Adhering to the FAIR Guiding Principles for scientific data management and stewardship, each statement can be explicitly traced back to the original paragraph and article in which it appears.

4.3. Human Annotation Process

For the human annotation process, we developed detailed annotation guidelines, and we employed three annotators. For each step, annotators first completed their annotations independently and then consolidated their results, discussing any disagreements until reaching consensus. We assessed the annotation reliability using Krippendorff's α, Cohen's κ, and the % agreement.

4.3.1. Human Annotation of Statement Extraction

For each LLM-proposed statement, annotators judged whether it was acceptable (*ok*) or problematic (*problem*), following the annotation guidelines requiring that each statement be:

- **Atomic**: expressing a single verifiable proposition.

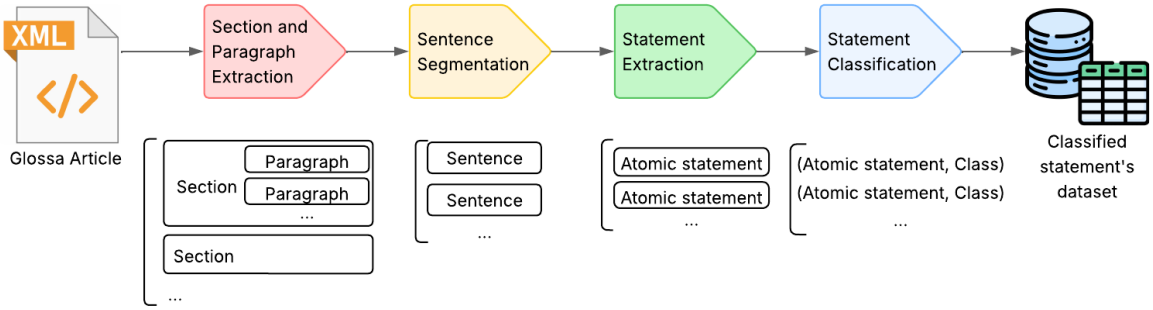

Figure 1: Pipeline used for extracting and classifying atomic statements from XML-encoded articles. The top row shows processing steps, while the bottom row shows the corresponding intermediate representations.

- **Independently verifiable**: understandable without the source paragraph.
- **Inferable from the source text**: not introducing information beyond the paragraph.

Metric	Value
Krippendorff's α (3 annotators)	0.299
Cohen's κ (Ann. 1 & Ann. 2)	0.397
Cohen's κ (Ann. 1 & Ann. 3)	0.208
Cohen's κ (Ann. 2 & Ann. 3)	0.301
% agreement (Ann. 1 & Ann. 2)	81.3%
% agreement (Ann. 1 & Ann. 3)	73.1%
% agreement (Ann. 2 & Ann. 3)	78.4%

Table 1: Inter-annotator agreement for Statement Extraction between the three annotators.

Table 1 reports the inter-annotator agreement. The relatively low values ($\alpha = 0.299$) reflect the inherent difficulty of the task. Although raw percentage agreement is comparatively higher (73.1%–81.3%), this likely overestimates true agreement due to chance effects and the uneven distribution between the *ok* and *problem* categories.

The annotators reported several challenges during the annotation process. First, the scope of an in-text citation can be ambiguous, especially when the citation is preceded or followed by multiple claims. Although annotators were asked to err on the side of caution and select the smaller scope when in doubt, the difference in judgment on scope remained a considerable source of disagreement. Furthermore, there can be significant subjective variation in defining what constitutes an atomic statement, particularly when there is ambiguity in coordination scope. The disagreement between annotators mostly did not involve the content of the claims themselves, but rather alternative interpretations of scope and atomicity. In the vast majority of cases, these interpretations may all arguably serve as gold statements without affecting their veracity value. Based on these observations, the annotation guidelines were subsequently refined.

Following annotation, the three annotators held a discussion meeting to reconcile disagreements and produced **342 human-curated gold statements** extracted from the 63 paragraphs.

4.3.2. Human Annotation of Statement Classification

The same three annotators independently classified the curated statements into the categories defined in Section 3.2. For statements labelled as L-Spec, they were also asked to provide the language referred to in each statement. Agreement was substantially higher than for the extraction task (Table 2).

Metric	Value
Krippendorff's α (3 annotators)	0.750
Cohen's κ (Ann. 2 & Ann. 1)	0.725
Cohen's κ (Ann. 2 & Ann. 3)	0.825
Cohen's κ (Ann. 1 & Ann. 3)	0.700
% agreement (Ann. 2 & Ann. 1)	81.4%
% agreement (Ann. 2 & Ann. 3)	87.6%
% agreement (Ann. 1 & Ann. 3)	79.4%

Table 2: Inter-annotator agreement for Statement Classification between the three annotators.

The most frequently identified languages in the L-Spec statements were: English, North Sámi, Yorùbá, Spanish, Basque, Polish, German, and Russian. Other languages appearing in the annotated dataset (each with 1 or 2 statements) include Mauritian Creole, Greek, Bulgarian, Saudi

Arabic, Brazilian Portuguese, Indonesian, Warlpiri, Samoan, Niuean, and Japanese. Different annotators showed high agreement on these language assignments, with English and North Sámi consistently appearing as the most common subjects of study. The low number of language-specific tags relative to the total number of statements indicates that the dataset primarily consists of general linguistic claims (L-Theo or L-Spec) rather than specific language data points (L-Typo).

A gold classification for all 342 statements was produced based on the individual annotations. If the three humans agreed, their label was kept; disagreements were resolved by manual review. The final dataset contains: 115 L-Spec, 110 L-Theo, 10 L-Typo, 91 C, and 15 S.

5. Experimental Setup

Evaluated Models. We ran the evaluation on eight LLMs spanning four provider families: GPT-5.2, GPT-5 Mini, and GPT-5 Nano (OpenAI); Gemini 3 Flash and Gemini 2.5 Flash (Google); DeepSeek V3.1 (DeepSeek/Fireworks); and Kimi K2.5 and Kimi K2 Instruct (Moonshot/Fireworks). The selection includes both open- and closed-source models with varying capabilities, enabling a more comprehensive evaluation.

Prompts settings. The models were evaluated using zero-shot prompts and tasked with returning only JSON arrays to facilitate post-processing of the results. The complete text of the prompts used is in Appendix A.

Challenge Dataset for Task 3. To evaluate whether LLMs possess genuine linguistic knowledge, we focused on the 235 L-type statements—115 L-Spec, 110 L-Theo, 10 L-Typo—since the S statements do not make a verifiable scientific claim, and the authors do not own the C statements. We constructed a challenge set of true and false statements. The 235 linguistic gold statements served as the true set. For each true statement, we generated two false variants: one using GPT-5.2 and one using Gemini 2.5 Flash. We used a contradiction prompt that requires:

> *Write a statement that is a strict logical contradiction of the claim below: both statements cannot be true at the same time. Keep the wording and structure as close as possible. The change must reverse the truth conditions of the claim. Do not use "no", "not", or explicit negation. Keep about the same length.*

Human annotators reviewed the automatically generated false statements to ensure that they contradicted the original version. The quality was generally very high and only a few statements required revision.

6. Results and Discussion

6.1. Task 1: Statement Extraction

We evaluated how well different LLMs extract statements by comparing their predictions against the 342 gold statements. Given the paraphrastic nature of the task—a model may express the same fact in different words—we use a semantic similarity approach rather than an exact match.

Evaluation metrics. We embedded both gold and predicted statements using the `all-MiniLM-L6-v2` model from the Sentence-Transformers library (Reimers and Gurevych, 2019), chosen for its strong performance on short-sentence similarity and paraphrase detection. Following Hanna and Bojar (2021), we computed the cosine similarity matrix between gold and predicted embeddings. **Recall** is the mean of the maximum similarity for each gold statement (how well does the model cover the gold?). **Precision** is the mean of the maximum similarity for each predicted statement (how relevant are the predictions?). **F1** is the harmonic mean of precision and recall.

Model	Prec.	Rec.	F1
Gemini 3 Flash	0.907	**0.883**	**0.894**
GPT-5.2	0.906	0.871	0.888
GPT-5 Mini	0.916	0.856	0.885
Gemini 2.5 Flash	0.917	0.853	0.884
Kimi K2.5	0.904	0.854	0.878
Kimi K2 Instruct	0.911	0.772	0.836
GPT-5 Nano	0.873	0.743	0.803
DeepSeek V3.1	**0.965**	0.494	0.653

Table 3: Statement Extraction evaluation.

Results. Table 3 presents the performance of all evaluated models on task 1. Most models achieve F1 scores between 0.80 and 0.89. Gemini 3 Flash leads with an F1 of 0.894, closely followed by GPT-5.2 (0.888). Gemini 2.5 Flash and Kimi K2.5 also perform competitively, though slightly below the top tier. GPT-5 Nano, the smallest model tested, still achieves a respectable 0.803, suggesting that a strong performance on this task does not necessarily require large models. DeepSeek V3.1 is an outlier with the highest precision (0.965), but the lowest recall (0.494), indicating that it extracts very few, but highly accurate statements. In contrast, the best-performing models achieve a more balanced trade-off between precision and recall, resulting in higher overall effectiveness.

Discussion. The results highlight the importance of balancing precision and recall in Statement

Extraction. Models such as GPT-5.2 and Gemini 3 Flash achieve strong F1 scores by maintaining consistency across both metrics, rather than optimizing for one at the expense of the other. On the other hand, the behavior of DeepSeek V3.1 illustrates a precision-oriented strategy that may be useful in applications where false positives are particularly costly, but less suitable for comprehensive information extraction. Overall, these findings indicate that recent compact models (e.g., GPT-5 Mini) can rival or outperform larger systems.

A closer qualitative analysis revealed additional challenges beyond overall performance. In particular, resolving citation scope remains highly difficult for LLMs—consistent with the difficulties observed for human annotators in Section 4.3.1. Consider the sentence "Broadly speaking, agentive verbs usually occur with an ergative subject and the auxiliary edun 'have', whereas patientive verbs combine with an absolutive subject and the auxiliary izan 'be' (Levin 1983)" from Pineda and Berro (2020). In the human annotation, the citation is applied to both coordinated statements: the statement on agentive verbs and the statement on patientive verbs. However, the automatically extracted statement, which applies the citation only to the latter, is also a plausible interpretation.

6.2. Task 2: Statement Classification

We evaluated how well LLMs classify the 342 gold statements into the categories defined in Section 3.2.

Evaluation metrics. For each model, we compute macro-averaged precision, recall, and F1-score against the gold labels.

Model	Prec.	Rec.	F1
Gemini 3 Flash	**0.824**	**0.890**	**0.840**
GPT-5.2	0.789	0.883	0.811
DeepSeek V3.1	0.774	0.870	0.805
GPT-5 Nano	0.747	0.799	0.767
Gemini 2.5 Flash	0.715	0.846	0.750
Kimi K2 Instruct	0.724	0.788	0.744
Kimi K2.5	0.733	0.770	0.742
GPT-5 Mini	0.679	0.683	0.680

Table 4: Statement Classification evaluation.

Results. Table 4 reports the performance of all evaluated models on task 2. Gemini 3 Flash achieves the best overall performance (0.840), consistent with its strong extraction performance. GPT-5.2 (0.811) and DeepSeek V3.1 (0.805) follow closely. While Gemini 2.5 Flash and the Kimi models obtain competitive results, they remain below the top-performing systems. GPT-5 Nano again demonstrates strong efficiency, achieving an F1 of 0.767 despite its smaller size. In contrast, GPT-5 Mini records the lowest performance (0.680), indicating greater variability across models for this task. In terms of individual metrics, most models maintain a relatively balanced precision–recall profile, although some variation is observed.

Model	Acc.	Prec.	Rec.	F1
Gemini 3 Flash	0.718	0.566	0.783	**0.657**
Kimi K2.5	0.680	0.526	0.740	0.615
Kimi K2 Instruct	0.643	0.490	**0.804**	0.609
DeepSeek V3.1	0.646	0.492	0.762	0.598
Gemini 2.5 Flash	0.649	0.494	0.736	0.591
GPT-5.2	**0.750**	**0.684**	0.515	0.587
GPT-5 Mini	0.703	0.567	0.596	0.581
GPT-5 Nano	0.696	0.575	0.455	0.508

Table 5: Overall Linguistics Knowledge Evaluation.

Discussion. Unlike task 1, where several models achieve similar performance, classification has more divergence across systems. Top-performing models such as Gemini 3 Flash and GPT-5.2 maintain a strong balance between precision and recall, which is crucial for achieving robust macro F1 performance across classes.

6.3. Task 3: Linguistic Knowledge Evaluation

As stated in Section 5, we used a challenge dataset of 705 statements—235 gold statements and 470 LLM-generated negations.

Evaluation metrics. We used macro-averaged accuracy, precision, recall, and F1-score against the gold labels. Beyond the binary verdict, we also analyzed the self-assessed confidence score[3] of each model.

Results. Table 5 presents the overall results of the Linguistics Knowledge Evaluation. Gemini 3 Flash achieves the highest F1 score (0.657) alongside strong recall (0.783), indicating robust coverage of linguistic phenomena. Across the eight models, accuracy ranges from 64.3% (Kimi K2 Instruct) to 75% (GPT-5.2), a spread of 11 percentage points. Within model families, a scaling effect is visible for OpenAI: GPT-5.2 (75%) $>$ GPT-5 Mini (70.3%) $>$ GPT-5 Nano (69.6%). For the Kimi family, Kimi K2.5 maintains a more balanced precision–recall profile, whereas Kimi K2 Instruct favors higher recall (0.804) at the expense of precision.

We examined whether the confidence score is a reliable indicator of accuracy by analyzing GPT-5.2, the best-performing model. Figure 2 plots accuracy

[3]All API calls use the provider default temperature (1.0 for OpenAI, Gemini, and Fireworks), since we are interested in the models' calibrated confidence rather than deterministic output.

and false-statement detection precision as a function of the model's own confidence. Statements are grouped by confidence threshold: each point shows the metric computed on the subset of statements for which the model reported confidence $\geq t$. The relationship is monotonic and substantial: at confidence ≥ 0.9, GPT-5.2 achieves 92.7% accuracy and 97.2% precision for false-statement detection ($n = 55$), compared with 73.4% and 83.0% over all 705 statements. A point-biserial correlation confirms the pattern (r=0.13, p<0.001); when restricted to true statements, the correlation between confidence and correct acceptance rises to r=0.39 (p<10^{-5}). Gemini 3 Flash, by contrast, reports confidence ≥ 0.8 for all the statements (mean = 0.93), leaving almost no room for threshold-based filtering. Its confidence score is therefore less informative as a triage signal, despite the model's competitive overall accuracy (71.8%).

A more extensive analysis of the models' performance by linguistic statement class is included in Table 6. Six out of eight models perform best on theoretical statements (L-Theo). Language-specific statements (L-Spec) are the most challenging, likely because they require knowledge of particular language data that may be underrepresented in training corpora. The L-Theo advantage over L-Spec is statistically significant for most models (chi-squared test: p<0.01 for Gemini 3 Flash, GPT-5 Mini, GPT-5 Nano, and Kimi K2.5; p<0.05 for Gemini 2.5 Flash and Kimi K2 Instruct) but not for GPT-5.2 (p=0.15) or DeepSeek V3.1 (p=0.12), suggesting that the best and worst models are less affected by statement type. Typological statements (L-Typo, n=30) show high variance due to the small sample size. The overall error rate (26–38%) indicates that current LLMs have non-trivial but insufficient linguistic knowledge for reliable fact-checking, reinforcing the need for human oversight or retrieval-augmented approaches.

Discussion. The results reveal a clear distinction between high-recall and high-precision modeling strategies. Models such as Gemini 3 Flash and Kimi variants prioritize recall, suggesting they are better suited for tasks requiring broad linguistic coverage, such as error detection or phenomenon identification. However, this often comes with reduced precision, potentially introducing more false positives. GPT-5 models exhibit a precision-oriented behavior, which may be advantageous in applications where correctness is critical (e.g., annotation pipelines), but their lower recall indicates limited coverage of diverse linguistic cases.

6.4. Tasks Comparison

Figure 3 highlights substantial heterogeneity across the three evaluation dimensions.

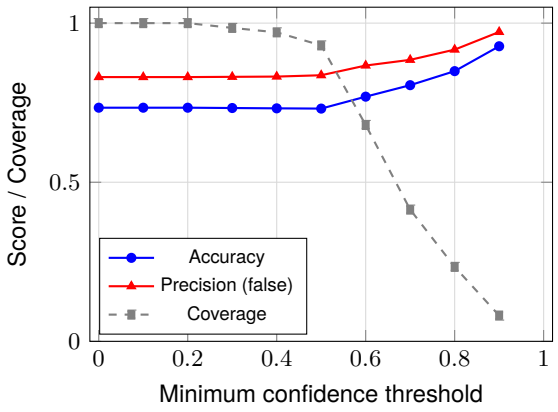

Figure 2: GPT-5.2 veracity performance as a function of minimum confidence threshold.

Model	Class	Acc.	Prec.	Rec.	F1
Gem. 3	L-Spec	0.67	0.51	0.76	**0.61**
	L-Theo	**0.78**	0.64	**0.80**	**0.71**
	L-Typo	0.67	0.50	0.90	0.64
Kimi k2.5	L-Spec	0.63	0.48	0.72	0.57
	L-Theo	0.74	0.60	0.75	0.67
	L-Typo	0.57	0.42	0.80	0.55
Kimi k2	L-Spec	0.61	0.46	**0.79**	0.58
	L-Theo	0.69	0.53	**0.80**	0.64
	L-Typo	0.60	0.45	**1.00**	0.62
DS V3.1	L-Spec	0.62	0.47	0.73	0.57
	L-Theo	0.68	0.52	0.78	0.63
	L-Typo	0.60	0.45	0.90	0.60
Gem 2.5	L-Spec	0.58	0.43	0.67	0.52
	L-Theo	0.73	0.58	**0.80**	0.67
	L-Typo	0.60	0.44	0.80	0.57
GPT-5.2	L-Spec	**0.73**	**0.64**	0.47	0.54
	L-Theo	0.77	**0.73**	0.55	0.63
	L-Typo	**0.77**	**0.64**	0.70	0.67
GPT-5m	L-Spec	0.65	0.50	0.53	0.51
	L-Theo	0.75	0.64	0.65	0.64
	L-Typo	**0.77**	0.62	0.80	**0.70**
GPT-5n	L-Spec	0.63	0.45	0.35	0.39
	L-Theo	0.76	0.70	0.55	0.62
	L-Typo	0.73	0.60	0.60	0.60

Table 6: Model performance on Linguistics Knowledge Evaluation by class.

Gemini 3 Flash is the strongest model overall, with the highest scores on the three tasks(Extraction 0.894, Classification 0.840, Linguistic Knowledge Evaluation 0.657).

The remaining models exhibit a more asymmetric behavior. GPT-5.2 is consistently high on Extraction and Classification, but weaker on Linguistic Knowledge Evaluation. DeepSeek V3.1 performs

substantially better on classification than on extraction, indicating that its conservative prediction strategy may be better suited to labeling predefined statements than identifying them in free text. Kimi variants are comparatively stronger on Linguistic knowledge than most models.

Overall, the results suggest that no single model family dominates uniformly across all tasks except Gemini 3 Flash. This reinforces the importance of evaluating models across multiple subtasks when assessing their suitability for complex information extraction pipelines.

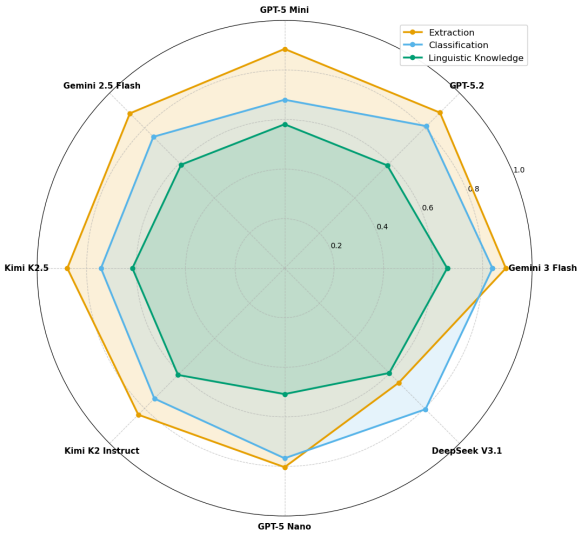

Figure 3: Overal F1-scores across the three tasks.

7. Conclusion

We have presented a comprehensive pipeline for extracting, classifying, and verifying linguistic claims from scientific articles. Our main findings are:

1. **Statement Extraction**: Gemini 3 Flash achieves the best F1 (0.894) for extracting atomic linguistic statements, with most models scoring above 0.80.

2. **Statement Classification**: Gemini 3 Flash again leads (F1 = 0.840), and inter-annotator agreement for classification ($\alpha = 0.750$) is substantially higher than for extraction ($\alpha = 0.299$).

3. **Linguistic knowledge Evaluation**: Current LLMs have limited but non-trivial linguistic knowledge. On a challenge set of 705 statements evaluated with eight models, the best accuracy is 75% (GPT-5.2), with theoretical statements being easier than language-specific ones. However, the top four models (accuracy 69–75%) are statistically indistinguishable (McNemar's test, all $p > 0.13$). Models range from conservative (GPT-5 Nano, high precision, low recall) to permissive (Kimi K2 Instruct, high recall, low precision), with GPT-5.2 achieving the best accuracy–recall trade-off. Within model families, larger models consistently outperform smaller ones.

We expanded this work to the rest of the paragraphs in the 11 Glossa articles. From the 1,082 unique paragraphs, we ended up with 3,655 unique sentences. A total of 7,713 atomic statements were extracted from the sentences, averaging 2.11 statements per sentence. All of the 7,713 atomic statements have been classified into L-Spec, L-Typo, L-Theo, C, or S.

Future work will extend the Linguistic Knowledge Evaluation to additional models and LLM-as-a-judge approaches. Another promising direction is systematic prompt optimization: all prompts in the current pipeline were manually designed, and frameworks such as DSPy (Khattab et al., 2023) could be used to automatically compile and tune prompt chains, potentially improving both extraction and veracity performance.

While the current collection already enables a first systematic evaluation, scaling it to several hundred thousand atomic statements annotated for L-Spec, L-Typo, and L-Theo would enable a more detailed and statistically reliable assessment of LLMs' linguistic knowledge. In particular, expanding the dataset to cover a broader and more diverse set of languages would allow us to test an important hypothesis: that LLMs capture the structural properties of well-resourced languages far better than those of typologically rare or low-resource languages. Such a resource would also provide the foundation for a standardized benchmark for linguistic knowledge evaluation, enabling systematic and reproducible comparison across models, prompting strategies, and future generations of LLMs.

Our long-term research goal is to incorporate corpus and treebank-based verification for L-Spec statements in UD-covered languages and investigate whether providing LLMs with retrieved context (e.g., relevant treebank data) improves their performance. Equally, we plan to extend our work on more typology-oriented journals, where we will verify with established typological databases such as WALS, Grambank, and APiCS.

8. Bibliographical References

Georgios Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2021. A review on fact extraction and verification. *ACM Computing Surveys*, 55(1):1–35.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: Why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93, Denver, CO. Association for Computational Linguistics.

Catherine Blake. 2010. Beyond genes, proteins, and abstracts: Identifying scientific claims from full-text biomedical articles. *Journal of Biomedical Informatics*, 43(2):173–189.

Clément Bleuze. 2024. *Analysing Claims in NLP Research*. Ph.D. thesis, INRIA.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1).

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2022. CS-KG: A large-scale knowledge graph of research entities and claims in computer science. In *International Semantic Web Conference*. Springer.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Tobias Jansen and Tobias Kuhn. 2016. Extracting core claims from scientific articles. In *Benelux Conference on Artificial Intelligence*, pages 33–46. Springer.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. 2023. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles: A survey. *Scientometrics*, 117:1931–1990.

Anna Pineda and Ane Berro. 2020. Hybrid intransitives in basque. *Glossa: a journal of general linguistics*, 5(1):22.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103. Association for Computational Linguistics.

Akshith Reddy Putta, Jacob Devasier, and Chengkai Li. 2025. ClaimCheck: Automatic fact-checking of textual claims using web evidence. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 303–316, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mourad Sarrouti, Asma Ben Abacha, Yassine M'rabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational*

Linguistics, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2023. Scientific fact-checking: A survey of resources and approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Wu Youyou, Yang Yang, and Brian Uzzi. 2023. A discipline-wide investigation of the replicability of psychology papers over the past two decades. *Proceedings of the National Academy of Sciences*, 120(6).

A. Prompts

A.1. Prompts used for Sentence Extraction

A.1.1. Paragraph segmentation

You are an expert assistant helping a linguist segment a scientific linguistics article into sentences.

Task: Split each numbered paragraph below into grammatically complete sentences. Preserve the original text exactly—do not paraphrase, correct, or reorder. Process each paragraph independently.

Rules: 1. Complete sentences only: Each output unit must be a grammatically complete sentence. Merge fragments (dangling clauses, isolated citations, orphan parentheticals) with the sentence they belong to. 2. Linguistic examples: Numbered examples (e.g., "(1)", "(2a)") must be attached to the sentence that introduces or discusses them *without* their glosses/translations. Keep only the original text, removing gloss lines and translation. Input: "Example (1) shows the problem in Spanish. (1) El niño duerme. 'The boy sleeps.'" Output: ["The Example "El niño duerme." shows the problem in Spanish: "] Linguistic examples that are not cited or attached to any sentence must be completely removed. 3. In-text citations: Keep citations attached to their sentence. 4. Lists and enumerations: If a sentence introduces a list, keep short list items with the introducing sentence. Split only if items are full sentences themselves. 5. Section titles: Titles/headings may remain as standalone fragments (the only exception to rule 1). 6. Preserve numbering: Only include numbers if the original text has them.

A.1.2. Statement Extraction

You are an expert linguist extracting verifiable atomic statements from a linguistics article. Process each paragraph block below independently.

Core rules — each extracted statement must: 1. Be atomic: split coordinations ("A and B") into separate statements when each conjunct is independently meaningful. When atomicity is ambiguous, do not split. Exception: joint reference lists stay together (e.g., "Smith (2020) and Miller (2023) show that. . . "). 2. Be a factual assertion that can be independently understood and assessed. 3. Be directly and literally inferable from the source text — no chained inferences (from "A then B" and "B then C", extract only those two, never "A then C"). 4. Be self-contained: resolve all pronouns/references (e.g., "it" then "the subject NP") and expand abbreviations (e.g., "NP" then "noun phrase (NP)"). 5. Preserve technical precision. Always include the language name when the context makes it clear (e.g., "The verb agrees. . . " then "In Spanish, the verb agrees. . . "). 6. Fold linguistic examples into the statement text; remove example numbers, glosses, and translations, keeping only the original-language form. 7. Pure conditional ("if A then B"): extract only the whole implication as one statement; do not extract A or B separately. 8. Causal/inferential ("A. Hence B"): extract (a) A as a standalone statement, and (b) the implication "A implies B". Do NOT extract B alone. - Source: "Prepositions act as cues to retrieve the PP correlate. Hence, P-omission in sluices comes with a processing cost." - (a) "Prepositions act as cues to retrieve the PP correlate." (b) "Prepositions acting as cues to retrieve the PP correlate implies that P-omission in sluices comes with a processing cost." 9. If the author attributes a claim to another work, keep the attribution. If the author also endorses the claim, additionally extract the general (non-attributed) version (e.g., "X (2024) uses Y to prove Z" → also "Y can be used to prove Z"). 10. Ambiguous citation scope: assume the smallest scope (only the immediately preceding clause). Text after the citation is the current author's claim unless marked otherwise. 11. Quoted cited text: preserve verbatim; do not split further. 11. "X followed Y in their analysis of Z" then (a) "Y analyzes Z" and (b) "X uses the same methodology as Y for Z". 12. Non-restrictive "which" clauses yield an additional statement (e.g., "French, which is Indo-European, . . . " becomes "French is an Indo-European language."). If restrictiveness is uncertain, do not split.

A.2. Prompt used for Statement Classification

You are an expert linguist reviewer classifying statements from a linguistics article according to specific guidelines.

Statements to classify: {paragraph blocks}

Categories: * C (Citation/Attribution): State-

ments attributing an idea or data to another author or paper. Example: "As noted by Chomsky (1995), the minimal link condition applies." Ambiguity: If the statement is "according to" a paper or person, it is Type C. But if it is "according to" a rule or principle (even if attributed earlier), it is likely Type L (author asserting the rule's validity). Scope: "Ginzburg & Sag (2000) proposed [X]" is C. But the continuation "an interrogative sluice is not derived..." is L (if asserted by the current authors).

* L (Linguistic Statement): The authors themselves assert the claim. Sub-types:

- L-Spec (Linguistic - Specific): Empirical claims about a *specific* language. MUST NOT make comparisons with other languages. Verifiable on language data (corpora, treebanks). Language Field: You MUST specify the language name for L-Spec statements. Example: "Warlpiri is mostly SOV." (Language: Warlpiri) - L-Typo (Linguistic - Typological): Cross-linguistic phenomena or comparisons. Describes phenomena across languages ("All VSO languages have prepositions"). Compares a set of languages ("Sino-Tibetan languages tend to be more..."). Compares specific languages ("Polish is similar to Russian..."). - L-Theo (Linguistic - Theoretical): Theory-internal generalizations or claims. Not directly verifiable with raw language data (treebanks, corpora). Abstract concepts (e.g., "Move-α is blocked by locality constraints"). Evaluating theories ("This mismatch casts doubt on syntactic theories...").

* S (Structural/Methodological): Statements that do not make a scientific claim. Section titles ("3.2 Theoretical Background"). Definitions ("We define <term> as..."). Meta-discourse/Signposting ("Section X will review...", "The authors studied...", "This article"). Data information ("Our dataset contains 1200 sentences"). Methodology ("We use the Fisher test...", "We asked 12 native speakers..."). Technical/Algorithmic results ("It took 12 seconds", "20% were rejected").

Important Guidelines for Ambiguity: 1. If there is a citation in the statement -> Type C. (e.g., "Many non-P-stranding languages allow P-omission (Fortin 2007)..." -> Type C). 2. "According to [Rule/Principle]" -> Type L. 3. "According to [Person/Paper]" -> Type C.

Instructions: For each statement, provide: - Primary 'class' (L-Spec, L-Typo, L-Theo, C, S) and 'confidence' (0-1). - If L-Spec, provide the 'language' (e.g., "French", "Warlpiri"). - Dictionary/standard name. If not L-Spec, use null. - 'secondary_class' and 'secondary_confidence'. - Keep the 'statement_idx'.

A.3. Prompts used for Linguistic Knowledge Evaluation

A.3.1. Statement Negation

You are a researcher writing a paper. For each statement, write a statement that is a STRICT logical contradiction of the claim below: both statements cannot be true at the same time.

Rules: For each statement, - Keep the wording and structure as close as possible. - The change must reverse the truth conditions of the claim (not just express a different opinion). - Do not use "no", "not", or explicit negation. - Keep about the same length. Return a JSON object with a single key "contradiction".

A.3.2. Linguistic Knowledge Evaluation

Role: Linguistic Fact-Checker (Syntax/Morphosyntax). Task: Evaluate the accuracy of STATEMENT based on descriptive academic linguistics.

RULES: - No prescriptive grammar; use descriptive evidence. - If the statement is a single word or lacks propositional content, veredict is false. - Lower confidence for theoretical disagreements. - Return exactly one JSON object per statement, in order.