FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGY CENTER
DEPARTMENT OF INFORMATICS AND STATISTICS
UNDERGRADUATE PROGRAM IN INFORMATION SYSTEMS

Gabriel Avila

A DUAL-STAGE DETECTION PIPELINE FOR METAPHOR ANNOTATION:
Using LLM-Based Proposal Generation and Linguistic Verification

Florianopolis, Santa Catarina – Brazil
2026

Gabriel Avila

A DUAL-STAGE DETECTION PIPELINE FOR METAPHOR ANNOTATION:
Using LLM-Based Proposal Generation and Linguistic Verification

Course Completion Paper submitted to the Undergraduate Program in Information Systems of the Technological Center of the Federal University of Santa Catarina as a requirement for obtaining the degree of Bachelor in Information Systems.
Supervisor: Prof. Rodrigo Bragio Bonaldo, Dr.
Co-supervisor: Prof. Jean Carlo Rossa Hauck, Dr.

Florianopolis
2026

Avila, Gabriel
A Dual-Stage Detection Pipeline for Metaphor Annotation: LLM-Based Proposal Generation and Linguistic Verification / Gabriel Avila ; orientador, Rodrigo Bragio Bonaldo, coorientador, Jean Carlo Rossa Hauck, 2026.
43 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Sistemas de Informação, Florianópolis, 2026.

Inclui referências.

1. Sistemas de Informação. 2. Metaphor annotation. 3. Semantic preference violation. 4. Metaphor identification. 5. Linguistic annotation. I. Bragio Bonaldo, Rodrigo. II. Rossa Hauck, Jean Carlo. III. Universidade Federal de Santa Catarina. Graduação em Sistemas de Informação. IV. Título.

Gabriel Avila

A DUAL-STAGE DETECTION PIPELINE FOR METAPHOR ANNOTATION:
Using LLM-Based Proposal Generation and Linguistic Verification

This Course Completion Paper was considered adequate for obtaining the title of Bachelor in Information Systems and approved in its final form by the Undergraduate Program in Information Systems.

Florianopolis, July 10, 2026.

Prof. Álvaro Junior Pereira Franco, Dr.
Course Coordinator

Examining Board:

Prof. Rodrigo Bragio Bonaldo, Dr.
Supervisor
UFSC/CFH/HIS

Prof. Jean Carlo Rossa Hauck, Dr.
Co-Supervisor
UFSC/CTC/INE

Rodrigo Souza Wilkens, Dr.
Evaluator
UCLouvain / SSH / IL&C

This work is dedicated to my dear parents.

ACKNOWLEDGMENTS

I would like to thank Stefan Curi for his guidance and support throughout this thesis. His advice, feedback on my code, and the discussions we had were very helpful and greatly improved the quality of my work. I appreciate the time he dedicated to supervising this project.

I also thank the members of the examination board for their careful reading and constructive suggestions, which helped improve the clarity and completeness of this work.

DECLARATION OF AI USE

Generative artificial intelligence tools were used as support tools during the preparation of this thesis. OpenAI ChatGPT, including web-search assistance, was used to help map terminology, identify candidate references and adjacent research directions, review wording, check consistency, and support document editing.

The latest OpenAI Codex model available through the terminal environment was used as an execution assistant for thesis polishing, Typst editing, Mermaid diagram generation, exploratory code prototypes, implementation work, and compilation checks. Some Codex-assisted prototypes were discarded and are not part of the final system. For the code that remains, Codex acted as an executor under author direction: design decisions, acceptance criteria, interpretation of results, and final inclusion decisions were made by the author. During development, author guidance also included code-organization preferences such as favoring functional composition over object-oriented structure where practical and avoiding unnecessary helper abstractions. These preferences informed the implementation process but were not treated as independent evaluation criteria for the final thesis. Code quality was controlled through the project's development workflow, including Ruff-based formatting and linting, Pyrefly type checking, project tests, and manual review.

The author reviewed, selected, rewrote, and validated the final content. The tools were not treated as authors and were not used as a substitute for the implementation, empirical evaluation, interpretation of results, or final academic responsibility for the work. All claims, citations, code, benchmark results, and conclusions remain the responsibility of the author.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

DMD	Dual-Perspective Metaphor Detection system
LLM	Large Language Model
MIP	Metaphor Identification Procedure
MIPVU	Extended Metaphor Identification Procedure
MLM	Masked Language Model
NLP	Natural Language Processing
POS	Part-of-Speech
SPV	Semantic Preference Violation

CONTENTS

ABSTRACT

Metaphor annotation is useful for linguistics, discourse analysis, digital humanities, and Natural Language Processing (NLP), but manual identification requires specialized knowledge and does not scale easily to large corpora. This thesis presents `metaphoriq-detector`, a dual-stage metaphor detection pipeline that combines Large Language Model (LLM) based proposal generation with deterministic linguistic verification. The proposal stage identifies candidate metaphorical spans and produces structured MIP/SPV-based explanations. The verification stage checks those proposals using lexical, syntactic, and semantic evidence, including part-of-speech analysis, WordNet-derived basic-contextual contrast proxies, and semantic-preference signals. The detector was evaluated on VUA20, TroFi, and MOH-X against XLM-R, CreativeLang RoBERTa, FrameBERT, and MelBERT. In this benchmark, `metaphoriq-detector` did not surpass the strongest academic baseline, but it showed a high-recall profile with lower precision after an uncertainty-triggered retry step. The results indicate that separating LLM proposal generation from external verification yields an inspectable candidate-generation pipeline, while also exposing the precision limits of the implemented adjudication strategy.

Keywords: Metaphor annotation; Metaphor identification; Large language models; Linguistic annotation; Natural language processing; Semantic preference violation.

RESUMO

A anotação de metáforas é relevante para linguística, análise do discurso, humanidades digitais e NLP, mas a identificação manual exige conhecimento especializado e não escala facilmente para grandes corpora. Este trabalho apresenta `metaphoriq-detector`, um pipeline de detecção de metáforas em duas etapas que combina geração de propostas baseada em LLM com verificação linguística determinística. A etapa de proposta identifica possíveis trechos metafóricos e produz explicações estruturadas baseadas em MIP/SPV. A etapa de verificação avalia essas propostas por meio de evidências lexicais, sintáticas e semânticas, incluindo análise de classes gramaticais, proxies de contraste entre sentido básico e contextual derivados da WordNet e sinais de violação de preferência semântica. O detector foi avaliado nos conjuntos VUA20, TroFi e MOH-X em comparação com XLM-R, CreativeLang RoBERTa, FrameBERT e MelBERT. Nesse benchmark, `metaphoriq-detector` não superou o melhor baseline acadêmico, mas apresentou alta revocação e menor precisão após uma etapa de retentativa acionada por incerteza. Os resultados indicam que separar a geração de propostas por LLM da verificação externa produz um pipeline inspecionável de geração de candidatos, ao mesmo tempo em que expõe os limites de precisão da estratégia atual de adjudicação.

Palavras-chave: Anotação de metáforas; Identificação de metáforas; Modelos de linguagem de grande porte; Anotação linguística; Processamento de linguagem natural; Violação de preferência semântica.

1 INTRODUCTION

Metaphor annotation supports research in linguistics, discourse analysis, digital humanities, and NLP, but it remains difficult to scale. Manual procedures such as Metaphor Identification Procedure (MIP) and Extended Metaphor Identification Procedure (MIPVU) provide explicit criteria for deciding whether a lexical unit is metaphorical, yet they require specialized knowledge and sustained human effort (The Pragglejaz Group, 2007); (Steen et al., 2010). This limitation becomes more visible in historical corpora, where meanings may shift over time and annotation decisions must remain interpretable for later expert review (Tahmasebi; Borin; Jatowt, 2021).

Recent neural and LLM based approaches have improved automatic metaphor detection, but they do not fully solve the transparency problem. Systems such as MelBERT (Choi et al., 2021) and dual-perspective LLM frameworks (Lin et al., 2024) show that metaphor-theoretic signals can improve detection, while also revealing a remaining tension between predictive performance and auditability. In particular, approaches that depend on direct model judgment or self-verification may produce plausible decisions without exposing enough independent evidence for why a candidate expression should be accepted or rejected (Stechly; Valmeekam; Kambhampati, 2025).

This thesis addresses that gap through `metaphoriq-detector`, a dual-stage metaphor detection pipeline developed for annotation workflows in digital humanities and NLP. The system separates candidate generation from verification: an LLM proposes metaphorical spans and structured explanations (Chen; Yang; Huang, 2024), while a deterministic verification layer evaluates those proposals using lexical, syntactic, and semantic evidence inspired by MIP and semantic preference violation (Shutova; Teufel; Korhonen, 2013). This separation is intended to preserve the interpretive flexibility of LLMs without delegating the final decision entirely to model self-assessment.

The work is motivated by the activities of (Universidade Federal de Santa Catarina. IA e História: Grupo de Estudos e Pesquisa Interdisciplinar, 2025), which studies historical corpora and requires scalable support for identifying metaphorical language. Similar digital-humanities projects often need computational tools that adapt to corpus-specific characteristics while keeping the annotation process inspectable (Hengchen et al., 2024); (Herrmann; Reiter; Petris, 2023). For that reason, the thesis focuses not only on detection performance, but also on the structure of the decision process and the metadata produced during annotation.

The main contribution of this work is the design, implementation, and empirical evaluation of `metaphoriq-detector`. The evaluation compares the proposed pipeline with selected metaphor-detection baselines and analyzes its predictive performance.

Objectives

General Objective

Develop and evaluate a dual-stage pipeline for metaphor detection that combines LLM based proposal generation with linguistically informed verification, aiming to produce inspectable metaphor annotations for digital-humanities and NLP workflows.

Specific Objectives

- Establish the linguistic and computational criteria used by the detector, especially MIP, Semantic Preference Violation (SPV), span-level annotation, and the need for auditable evidence.
- Specify a detection architecture that separates LLM proposal generation, deterministic verification, and threshold-based adjudication.
- Implement `metaphoriq-detector` with structured JSON proposals, span sanitization, Part-of-Speech (POS)/dependency analysis, WordNet-based contrast proxies (Miller, 1995), and semantic-preference signals (Shutova; Teufel; Korhonen, 2013).
- Compare detector profiles through versioned benchmark experiments in order to select and analyze an operating point for academic evaluation.
- Evaluate the selected detector against XLM-R (Conneau et al., 2020), CreativeLang RoBERTa (CreativeLang, 2025), FrameBERT (Li et al., 2023), and MelBERT (Choi et al., 2021) using precision, recall, F1, accuracy, and confusion-matrix counts.

Organization of the Work

The thesis is organized as follows. Section 1.1 presents the linguistic and computational concepts that support the detector. Section 1.2 discusses prior metaphor-detection systems and positions this work in relation to neural, LLM based, and verification-oriented approaches. Chapter 2 describes the architecture and implementation of `metaphoriq-detector`. Chapter 3 presents the benchmark setup and quantitative results. Finally, Chapter 4 summarizes the findings, limitations, and future work.

1.1 THEORETICAL FOUNDATION

Automatic metaphor detection depends on a small set of linguistic and computational assumptions. This chapter introduces those assumptions: metaphor as a relation between domains (Lakoff; Johnson, 1980), the contrast between basic and contextual meaning (The Pragglejaz Group, 2007); (Steen et al., 2010), lexical resources for approximating word senses (Miller, 1995), and computational signals that can make detection decisions inspectable (Shutova; Teufel; Korhonen, 2013); (Choi et al., 2021).

1.1.1 Metaphor and Meaning

1.1.1.1 Conceptual Metaphor

The contemporary study of metaphor is strongly shaped by the cognitive account proposed by (Lakoff; Johnson, 1980). In this view, metaphor is not merely an ornamental use of language. It is a way of structuring one domain of experience through another. A concrete or familiar source domain is used to reason about a more abstract target domain.

Classic examples include ARGUMENT IS WAR and TIME IS MONEY. Expressions such as “defend your point” or “waste time” do not describe literal combat or financial exchange, but they borrow structure from those domains to organize abstract experience (Lakoff; Johnson, 1980). For annotation, this matters because the surface phrase alone is not enough: the annotator must ask how the expression is being used in context.

1.1.1.2 Basic and Contextual Meaning

Metaphor identification often starts from a contrast between a word’s basic meaning and its contextual meaning. The basic meaning is usually more concrete, bodily, historically earlier, or more directly tied to physical experience. The contextual meaning is the sense required by the sentence in which the word appears.

The verb “grasp” illustrates the distinction. In “grasp the handle”, the verb denotes physical holding. In “grasp the concept”, the physical action is no longer literal, but the abstract act of understanding is interpreted through it. This kind of contrast is central to Metaphor Identification Procedure (The Pragglejaz Group, 2007).

1.1.1.3 Lexical Senses

Computational systems need an approximation of those meanings. Lexical resources such as WordNet represent senses through synsets, grouping near-synonymous words and attaching glosses to each sense (Miller, 1995). These glosses can be used as reference points when comparing a word’s contextual use with a more basic or conventional sense.

This is only an approximation. The first WordNet synset of a lemma, for example, is often used as a proxy for a frequent or default sense, but it does not always correspond to the exact basic meaning an annotator would select. Still, lexical resources provide a reproducible source of evidence, which is useful when the goal is not only to classify a sentence but also to explain why the decision was made.

1.1.2 Linguistic Identification Procedures

1.1.2.1 The Metaphor Identification Procedure

The MIP gives annotators a step-by-step procedure for deciding whether lexical units are used metaphorically in discourse (The Pragglejaz Group, 2007). The annotator identifies the lexical unit, determines its contextual meaning, checks whether it has a more basic meaning, and then decides whether the contextual use can be understood through comparison with that basic sense.

If such a contrast exists and the contextual meaning can be related to the basic one by comparison, the use is marked as metaphorical. If the contextual meaning matches the basic meaning, or if no meaningful contrast is present, the use is treated as literal. The value of MIP is not that it removes judgment completely, but that it makes the judgment explicit.

1.1.2.2 The Extended Metaphor Identification Procedure Extension

MIPVU extends MIP with more detailed rules for multiword expressions, idiomatic uses, and dictionary consultation (Steen et al., 2010). These details are important because metaphor annotation often fails at the boundaries: deciding what counts as the unit under analysis can change the final label.

For this thesis, MIPVU is relevant mainly as a reminder that detection is a span-level task. A system should not only say whether a sentence is metaphorical; it should also preserve which expression triggered the decision.

1.1.2.3 Semantic Preference Violation

Semantic preference describes the tendency of words, especially verbs, to occur with arguments of certain semantic types. The verb “eat” normally selects something edible; “die” normally selects something living. When those expectations are violated, figurative interpretation becomes more likely.

In “the idea died”, the verb selects an abstract noun rather than a living entity. The mismatch does not prove metaphor by itself, but it is useful evidence. Computational metaphor research has used this kind of signal to identify semantic anomalies in verb-argument structures (Shutova; Teufel; Korhonen, 2013).

1.1.3 Computational Detection

Early computational approaches used hand-crafted linguistic features: POS tags, dependency patterns, concreteness measures, and animacy-like cues. These features attempted to operationalize the same contrasts that human procedures describe, especially mismatches between expected and observed usage.

Neural models changed the representation of meaning. Contextual embeddings allow the same word to receive different representations in different sentences, which helps capture sense shifts and non-literal uses. Architectures such as MelBERT use this contextual behavior while incorporating metaphor-identification theory into the model design (Choi et al., 2021).

LLMs add another possibility: they can produce structured explanations of a candidate metaphor when prompted with linguistic criteria. They can identify a span, describe a possible contextual meaning, and articulate a comparison with a basic sense (Lin et al., 2024); (Chen; Yang; Huang, 2024). This makes them useful for proposal generation, especially when the system needs an interpretable hypothesis rather than only a label.

However, generation and verification are different tasks. LLMs may reject their own correct outputs or accept weak explanations when asked to evaluate themselves, a limitation observed in broader studies of LLM self-verification (Stechly; Valmeekam; Kambhampati, 2025). For metaphor detection, this is especially relevant because the decision often depends on checking a proposed interpretation against external lexical or contextual evidence.

The theoretical position adopted in this thesis follows from that limitation: the LLM proposes, but the final decision should be constrained by independent verification signals. Basic-contextual contrast, lexical-sense evidence, dependency structure, and semantic preference violations provide the linguistic basis for that separation.

1.2 RELATED WORK

This chapter reviews work that directly shaped the design of `metaphoriq-detector`: contextual support for metaphor detection, metaphor-theoretic prompting, limits of LLM self-verification, and explanation generation (Hayashi; Sasaki, 2025); (Lin et al., 2024); (Stechly; Valmeekam; Kambhampati, 2025); (Chen; Yang; Huang, 2024). As stated in the Declaration of AI Use, ChatGPT with web search supported the initial mapping of terminology, recent candidate papers, and adjacent research directions. The final selection of references was made through manual searches in Google Scholar, arXiv, and the ACL Anthology.

1.2.1 Context and Metaphor Detection

Metaphor detection often depends on information that is not fully contained in a single sentence. (Hayashi; Sasaki, 2025) study this problem by adding short LLM-generated auxiliary sentences around the input. Their results show that even limited synthetic context can improve detection accuracy across benchmark datasets, especially in ambiguous or low-information cases.

`metaphoriq-detector` also treats context as useful, but it uses it differently. Instead of generating extra text to enrich the input representation, it retrieves neighboring

sentences from the source document when they are available. A small context window supports the first proposal, and an explicit disambiguation retry is used when the first adjudication remains uncertain, with a larger neighboring context window added when available.

1.2.2 Theory-Guided LLM Detection

(Lin et al., 2024) introduce Dual-Perspective Metaphor Detection system (DMD), a framework that combines implicit representation-level signals with explicit MIP and SPV instructions. The paper is relevant here because it shows that metaphor-theoretic reasoning can be surfaced directly in LLM prompts and can improve interpretability when compared with purely implicit neural baselines.

The main difference is where reliability is placed. (Lin et al., 2024) includes a self-judgment mechanism in which the LLM validates or revises its own output. In `metaphoriq-detector`, the LLM produces the initial hypothesis, while the verification step is external to the model and relies on dependency analysis, Masked Language Model (MLM) probabilities, WordNet evidence, and proxies inspired by MIP and SPV.

Recent evaluations make this separation more important. (Sanchez-Bayona; Agerri, 2025) show that LLM performance on metaphor interpretation can be driven by surface features such as lexical overlap and sentence length rather than by deep metaphorical understanding. (Reimann; Scheffler, 2025) report similar fragility in zero- and few-shot metaphor detection, including overgeneralization in extended metaphor settings. These findings support treating direct LLM outputs as hypotheses that need external checks.

1.2.3 Self-Verification and Explanations

The decision to separate proposal from verification is also motivated by work on LLM self-critique. (Stechly; Valmeekam; Kambhampati, 2025) show that iterative self-verification can degrade performance in reasoning and planning tasks, partly because LLM-based verifiers produce high false-negative rates. Although their study is not about metaphor detection, the result matters here: verifying a generated interpretation is not automatically easier than generating it.

Metaphor detection has also moved toward explanation-oriented outputs. (Chen; Yang; Huang, 2024) introduce a task in which models generate natural-language reasons for metaphor usage. These explanations can be useful to humans, but they remain free-form and are not independently checked. `metaphoriq-detector` keeps the benefit of explanation by asking for structured proposals, then evaluates them with separate linguistic evidence.

Recent work also questions whether ordinary detection scores are enough to establish metaphor understanding. (Yang et al., 2026) argue for aptness judgments as a cognitive probe for language models, showing that binary metaphor labels capture

only part of the phenomenon. This does not replace the benchmark used in this thesis, but it motivates interpreting F1 as a task-specific measure rather than as a complete account of metaphor comprehension.

Taken together, these works show that recent metaphor detection has moved in three useful directions: adding context, incorporating metaphor-identification theory, and producing explanations. The remaining gap is methodological rather than only predictive. Explanations and self-judgments are useful, but they are weaker for annotation workflows when they are produced and validated by the same model, or when they are not connected to explicit linguistic evidence.

`metaphoriq-detector` is positioned as an inspectable candidate-generation pipeline for that setting. It uses surrounding document context when available, treats the LLM output as a hypothesis, and then checks the proposed spans with separate lexical, syntactic, and semantic signals. This does not remove the need for benchmark evaluation, but it makes the decision process more suitable for later review than a single end-to-end label or an unchecked free-form explanation.

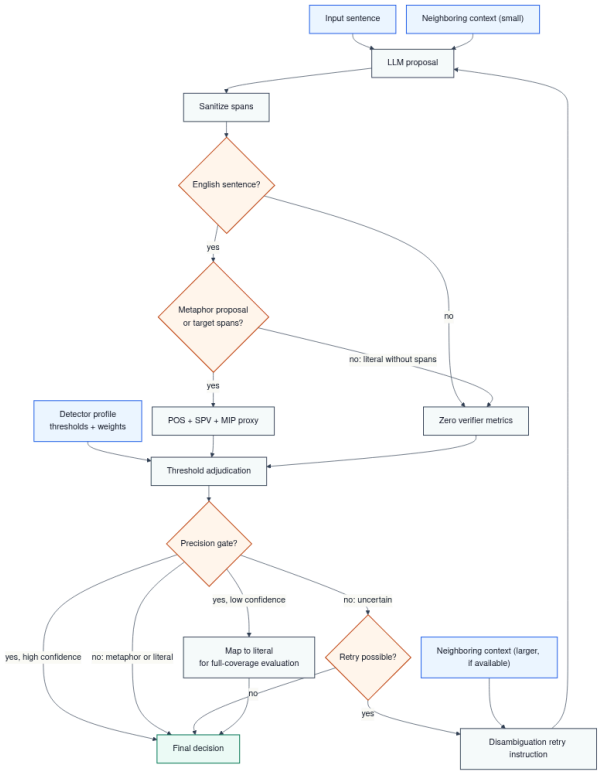
2 DEVELOPMENT METHODOLOGY

This chapter describes `metaphoriq-detector`, the metaphor-detection component evaluated in this work. The detector is organized around three stages: proposal generation, verification, and adjudication. The chapter focuses on those stages because they define the method evaluated in the following chapter.

2.1 SYSTEM OVERVIEW

`metaphoriq-detector` receives a sentence and builds a small neighboring context from the surrounding sentence on each side, when those neighbors are available. It asks an LLM to produce a structured metaphor proposal for the target sentence, using that context only for disambiguation. The proposal is then sanitized, optionally verified with linguistic evidence, and passed to an adjudication function that assigns the final label. When adjudication remains uncertain, the detector runs a second proposal pass with an explicit disambiguation instruction for the unresolved usage, using a larger neighboring context window when that context is available. The detector output contains the label, spans, explanation, verifier metrics, raw proposal, raw verification output, and decision metadata.

Figure 1: Dual-stage detection workflow

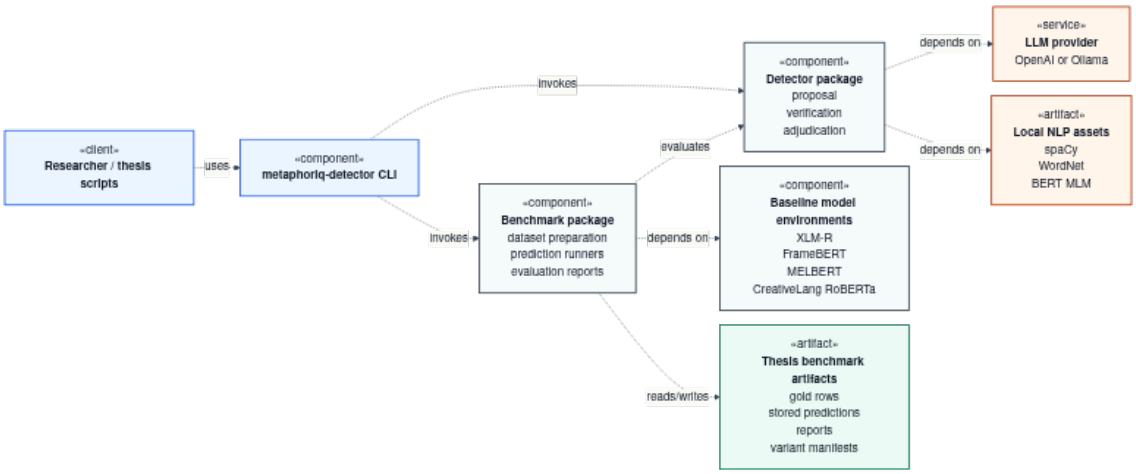


Source: Author

The workflow in Figure 1 describes the decision path inside the detector. The structural boundary of the implementation is shown in Figure 2. The detector package is kept separate from the benchmark package: benchmark code can call the detector

to produce predictions, but the detector does not depend on stored benchmark reports, historical experiment folders, or external baseline model environments. Those elements are part of the evaluation infrastructure rather than the detector itself.

Figure 2: UML-style standalone detector component diagram



Source: Author

2.2 PROPOSAL GENERATION

The proposal stage uses a structured prompt template rendered with the target sentence and, when available, disambiguating context or target information. On the retry path, the same proposal mechanism receives an additional instruction stating that the previous adjudication was inconclusive and asking the model to re-analyze the target usage as a binary metaphor/literal decision. The LLM call is routed through the project's JSON generation service using OpenAI's `gpt-5-nano` in the evaluated experiments.

The prompt instructs the model to follow MIP and SPV cues and to return strict JSON. The schema requires:

- `label`: either `metaphor` or `literal`;
- `spans`: copied from the sentence with `text`, `start`, and `end` offsets;
- `reason`: a structured object containing `mip`, `spv`, and `paraphrase_literal` fields.

Context is used only for disambiguation. The prompt explicitly requires all returned spans to come from the sentence itself, not from the auxiliary context. After the LLM returns a proposal, the implementation sanitizes spans by checking whether their offsets match the sentence text. If the offsets are wrong but the span text appears in the sentence, the detector repairs the offsets by text search; if the span cannot be found, it is dropped.

2.3 VERIFICATION

Verification is applied only when the sentence language is English and when the proposal needs verifier evidence. Literal proposals without spans skip the verifier and

receive zeroed verification metrics. Non-English sentences also skip the verifier and are handled through proposal-only adjudication.

For English sentences, verification combines three kinds of evidence. First, a syntactic analysis component is used for tokenization, part-of-speech tagging, and dependency analysis. Proposed spans are enriched with POS information from the parsed sentence.

Second, the detector computes a semantic-preference signal. It identifies candidate mask spans from dependency patterns such as verbs with subjects or objects, copular complements, nouns in "of" constructions, and nouns modified by adjectival or compound dependents. Each candidate is masked and scored with a masked language model (Devlin et al., 2019); low probability for the original token is treated as stronger anomaly evidence. The final SPV score is the maximum anomaly score across candidates.

Third, the detector computes a WordNet-based MIP proxy (Miller, 1995). For each proposed span, it selects a content-word head, retrieves candidate synsets, and compares the first synset gloss with the sentence context. Lower overlap between gloss words and contextual words increases the proxy score. This is not a full human MIP annotation, but a reproducible approximation of basic-contextual contrast.

The verifier returns a confidence score computed as a weighted combination of the two numeric proxies:

$$\text{confidence} = 0.55 \cdot \text{spv} + 0.45 \cdot \text{mip}$$

The verification output preserves the enriched spans, textual reason fields, and `_metrics` containing `spv_score`, `mip_any`, `mip_score`, and `confidence`.

2.4 ADJUDICATION

The adjudication step combines the proposal label with verifier confidence. In the evaluated configuration, English sentences are handled with fixed heuristic threshold rules:

- accept a metaphor proposal when confidence is at least `0.4`;
- accept a literal proposal when confidence is below `0.3`;
- force a metaphor decision when confidence is at least `0.7`, even if the proposal was literal;
- otherwise return `uncertain` and trigger the disambiguation retry.

These thresholds, as well as the `0.55` and `0.45` weights used in the confidence formula, were not optimized through an exhaustive parameter search or statistically calibrated decision procedure. They were arrived at through exploratory pilot runs on the 33-item-per-dataset screening setting described below. Candidate configurations were compared by their precision-recall behavior, and the retained values were the most useful operating point among the tested profiles. The values therefore define the

implemented detector behavior and should be treated as tunable design choices rather than theoretically derived constants. If the retry pass remains below the threshold rules, the detector uses the retry proposal label as the final binary decision. Uncertainty is therefore an internal state that triggers retry and is never returned as the detector's final label.

For non-English sentences, adjudication returns the proposal label and spans directly with decision mode `proposal_only`. This behavior avoids applying English-specific lexical evidence to languages for which the verifier is not designed.

2.5 ITERATIVE DETECTOR DEVELOPMENT

The detector was developed through a sequence of small, versioned benchmark experiments rather than through a single final implementation pass. Each experiment fixed a detector profile, ran that profile on the same academic benchmark selection, and stored the resulting predictions, metrics, detector configuration, and detector changes under a separate experiment directory. This made it possible to compare changes in prompt design, target-awareness, verification, and adjudication without overwriting previous runs.

The first reference point was a baseline snapshot of the existing detector outputs. Subsequent profiles changed one or two design choices at a time. The main profile families were:

- target-aware proposal generation, where benchmark target words are supplied to the prompt so that the model judges the relevant lexical usages rather than unrelated figurative language in the same sentence;
- literal-target verification, where target spans are still passed to the verifier even when the LLM proposal initially labels the sentence as literal;
- lower or higher metaphor-acceptance thresholds, used to explore the precision-recall tradeoff of the adjudication step;
- proposal-only and dual-proposal profiles, used to separate the contribution of the LLM proposal from the deterministic verifier;
- MIP-heavy and precision-gated profiles, used to test whether stricter lexical contrast evidence can reduce false positives;
- disambiguation-retry profiles, used to test whether `uncertain` adjudications should trigger a second prompt whose explicit goal is to resolve the target usage into a binary decision.

To keep iteration cost manageable, each candidate profile was screened with 33 sentences per academic dataset, for 99 benchmark items in total across VUA20 (Leong et al., 2020), TroFi (Birke; Sarkar, 2006), and MOH-X (Mohammad; Shutova; Turney, 2016). This pilot setting was used to choose promising detector profiles and to compare

precision-recall behavior across profile variants. Candidate configurations were ranked with pooled precision, recall, F1, and per-dataset F1.

The selected configuration was then evaluated with 333 sentences per academic dataset. The values 33 and 333 were feasibility-driven limits rather than statistically optimized sample sizes: the first allowed repeated profile screening, and the second was the largest uniform slice that could be processed for all datasets and baselines within the project constraints. Keeping the same number of items per dataset also prevented the pooled metrics from being dominated by a single source dataset. This larger evaluation produced the detector predictions used for comparison with the academic baselines. The two-stage procedure kept prompt and threshold iteration practical while still requiring the selected configuration to be evaluated on a larger, shared benchmark slice.

The iteration process also exposed an important methodological distinction between recall-oriented and precision-oriented detector behavior. Some profiles were able to recover nearly all metaphorical cases but also labeled many literal examples as metaphorical. Later profiles therefore treated the verifier not only as supporting evidence for positive metaphor decisions, but also as a gate that could reject low-confidence proposals. The evaluated configuration kept the recall-oriented thresholds but added an uncertainty-triggered disambiguation retry. The resulting evaluation still exposed a substantial false-positive problem.

2.6 CONTEXT RETRY

When context is available, the detector can use neighboring sentences from the same source document. A small context window uses one sentence on each side of the target sentence. A larger context window uses three sentences on each side.

The first proposal uses the small context window when available. If adjudication returns `uncertain`, and a larger context window is available, the detector retries proposal, verification, and adjudication with the larger context. There is no further retry after this second pass.

The same retry principle applies when no larger neighboring context is available. In that case, the detector keeps the same sentence and available target information, but still adds the explicit instruction explaining that the earlier adjudication was inconclusive and asking for the strongest binary judgment. If the verifier still leaves the decision below the threshold rules, the detector uses the retry proposal as the final binary label.

2.7 IMPLEMENTATION CONSTRAINTS

The main constraint is language support. Although the proposal stage can be used with any language supported by the configured LLM, the verifier is English-only in the implemented detector. As a result, non-English sentences do not receive WordNet-based MIP proxy evidence or SPV confidence evidence.

Another constraint is that the verifier uses proxies rather than full linguistic annotation. The MIP score depends on the first WordNet synset and gloss overlap, while the SPV score depends on masked-token probabilities over dependency-based candidates. These signals are useful for reproducible adjudication, but they are not equivalent to expert metaphor analysis.

Finally, the detector can still represent `uncertain` as an internal adjudication state. Uncertainty triggers the retry path, after which the detector returns either `metaphor` or `literal`; it is not exposed as a final label.

2.8 SOFTWARE ARTIFACT AVAILABILITY

The software artifact associated with this thesis is available in an online repository: `https://codigos.ufsc.br/avila.g/metaphoriq-detector#`. It includes the standalone detector, benchmark adapters, experiment scripts, and stored benchmark artifacts.

3 EVALUATION AND RESULTS

This chapter presents the empirical evaluation of the proposed metaphor detection pipeline, identified in the benchmark as Metaphoriq detector. The evaluation compares the proposed detector with established metaphor-detection baselines and reports aggregate benchmark performance. The results show that Metaphoriq detector is not the strongest system by pooled F1, but they also reveal a distinctive error profile: the detector reaches high recall while still producing many false positives.

The evaluation tests the complete dual-stage design introduced in the previous chapter against direct neural baselines. Unlike systems that return only a model prediction, Metaphoriq detector combines LLM proposal generation with deterministic linguistic verification, preserving intermediate evidence for later inspection.

3.1 BENCHMARK SETUP

The benchmark aggregates binary metaphor-detection results over three datasets: VUA20 (Leong et al., 2020), TroFi (Birke; Sarkar, 2006), and MOH-X (Mohammad; Shutova; Turney, 2016). Although some source datasets contain token-level or span-level annotations, the comparison reported here evaluates whether each benchmark item is predicted as metaphorical or literal. This level was chosen because it allows `metaphoriq-detector` and the selected baselines to be compared through the same pooled metrics.

These datasets were selected for comparability with the baseline systems rather than as a complete representation of metaphor use. They are public metaphor-detection resources that appear in the training, fine-tuning, or reported evaluation context of the models used in the benchmark, including MelBERT, FrameBERT, and the RoBERTa/ XLM-R token-classification baselines (Choi et al., 2021); (Li et al., 2023); (CreativeLang, 2025); (Conneau et al., 2020). Using this shared dataset family makes it possible to evaluate `metaphoriq-detector` against models whose expected operating conditions are tied to those resources.

The benchmark is therefore composed only of these three public datasets. In this evaluation, each benchmark item is represented as a single sentence; no neighboring source-document context is provided to the detector.

For each model, the evaluation records dataset-level F1 scores when available, pooled accuracy, pooled precision, pooled recall, pooled F1, macro F1, and confusion-matrix counts. The benchmark output also keeps the number of evaluated rows so that systems with non-binary outputs can be audited.

The evaluated systems include Metaphoriq detector and four comparison baselines: XLM-R (Conneau et al., 2020), CreativeLang RoBERTa (CreativeLang, 2025), FrameBERT (Li et al., 2023), and MelBERT (Choi et al., 2021). These baselines repre-

sent different neural approaches to metaphor detection and provide a reference point for assessing the strengths and limitations of the proposed dual-stage architecture.

3.2 BENCHMARK EXECUTION

Each evaluated system is connected to the benchmark through an adapter that normalizes its output into a shared prediction format: document identifier, model key, label, binary `is_metaphor` value, optional score, spans, and raw output. This normalization is necessary because the models do not expose predictions in the same way.

The XLM-R and CreativeLang RoBERTa baselines are executed as Hugging Face token-classification pipelines. Their token-level outputs are converted to a sentence-level metaphor decision: if at least one returned token group is marked with the metaphor label, the sentence is counted as metaphorical; otherwise it is counted as literal. FrameBERT is executed through its external repository and its tabular token output is grouped back into sentence predictions. Rows marked as borderline by FrameBERT are treated as `uncertain`, not as binary decisions. MelBERT is executed through its external checkpoint workflow, which requires target-token rows and then aggregates target predictions back to the sentence identifier.

`metaphoriq-detector` is adapted separately. When a benchmark row already contains an embedded detector result, that result is used directly. Otherwise, the detector is run on the benchmark sentence and its final label is converted to the same binary prediction format. In the evaluated detector configuration, an `uncertain` adjudication triggers a second disambiguation prompt before the benchmark label is produced. If the second adjudication remains uncertain, the retry proposal label becomes the detector's final binary decision.

Human-gold rows marked as uncertain are excluded from binary metrics and counted as skipped gold rows. When a model returns labels that cannot be mapped to `metaphor` or `literal`, those rows are kept out of binary precision, recall, and F1. In the final `metaphoriq-detector` run, the disambiguation retry produces a binary decision for every eligible item.

3.3 QUANTITATIVE RESULTS

Figure 3: Benchmark F1 comparison

Source: Author

Table 1: Benchmark metrics by evaluated model

Model	**Precision**	**Recall**	**F1**	**Accuracy**
Metaphoriq detector	0.5148	0.9630	0.6710	0.5395
XLM-R Hugging Face token baseline	0.5882	0.9589	0.7291	0.6527
CreativeLang RoBERTa token baseline	0.5905	0.9651	0.7327	0.6567
FrameBERT	0.5877	0.8481	0.6943	0.5903
MelBERT	0.6798	0.8850	0.7690	0.7407

Source: Author

The strongest pooled F1 is obtained by MelBERT, with a pooled F1 of 0.7690, precision of 0.6798, and recall of 0.8850. CreativeLang RoBERTa and XLM-R follow with pooled F1 scores of 0.7327 and 0.7291, respectively. Metaphoriq detector obtains a pooled F1 of 0.6710, below these three baselines and below FrameBERT's pooled F1 of 0.6943, although FrameBERT's coverage of 0.4324 limits direct comparison with full-coverage systems.

The main behavior of Metaphoriq detector is high recall rather than balanced classification. It reaches pooled recall of 0.9630, but its precision is only 0.5148. The confusion matrix shows 469 true positives and 442 false positives, with only 70 true negatives and 18 false negatives. This indicates that the detector is effective at recovering metaphorical cases, but remains too permissive when distinguishing literal abstract language from metaphorical usage. The disambiguation retry resolves uncertain cases into binary decisions, but the benchmark results still show that stricter calibration is necessary. The detector's high recall should therefore be read together with its high false-positive count.

3.4 DISCUSSION

The results characterize the evaluated configuration rather than the performance ceiling of the architecture. Metaphoriq detector identified 469 of the 487 metaphorical items, producing only 18 false negatives, but it also produced 442 false positives. MelBERT (Choi et al., 2021), CreativeLang RoBERTa (CreativeLang, 2025), and XLM-R (Conneau et al., 2020) therefore achieved a better balance between precision and recall. FrameBERT (Li et al., 2023) obtained a numerically higher F1, although its coverage of 0.4324 limits direct comparison with the full-coverage systems.

The observed imbalance should not be treated as an intrinsic recall-oriented property of the detector. Its confidence weights and adjudication thresholds were selected through exploratory experiments on a small screening sample, without systematic optimization on a separate development set. The current evaluation does not show whether the large number of false positives results primarily from weak verifier signals or from an inadequate decision boundary. It establishes that the present configuration is insufficiently calibrated, but not that the architecture cannot achieve a more competitive operating point.

This distinction matters because proposal generation, verification, and adjudication are independent components. The decision boundary and verifier evidence can be revised without replacing the proposal mechanism. Related systems show that metaphor-theoretic information can support competitive detection when it is learned or appropriately calibrated (Choi et al., 2021); (Li et al., 2023); (Lin et al., 2024). The next evaluation should therefore calibrate weights and thresholds on a disjoint development set, include stronger negative evidence for abstract literal language, and report sensitivity or precision-recall analyses. If these changes reduce false positives while preserving most of the current recall, the pipeline may approach or surpass the selected baselines. Establishing that result requires a new evaluation on untouched data.

4 CONCLUSIONS AND FUTURE WORK

4.1 CONCLUSIONS

This work presented `metaphoriq-detector`, a dual-stage pipeline for metaphor detection that combines LLM based proposal generation with deterministic linguistic verification. The system was designed to address a central limitation of direct neural or prompt-only approaches: they can identify plausible metaphorical uses, but their decisions are often difficult to audit and may rely on unstable self-verification (Stechly; Valmeekam; Kambhampati, 2025). By separating proposal generation from verification, the pipeline uses the interpretive capacity of LLMs while grounding the final decision in explicit lexical, syntactic, and semantic evidence. Its output therefore exposes spans, explanations, verification signals, and adjudication metadata rather than only a classifier score.

The empirical evaluation does not show that `metaphoriq-detector` outperforms the strongest selected baselines. MelBERT (Choi et al., 2021), CreativeLang RoBERTa (CreativeLang, 2025), and XLM-R (Conneau et al., 2020) obtain higher pooled F1 scores on the evaluated benchmark slice. Instead, the results characterize a provisionally calibrated operating point. They show that the evaluated configuration recovers most metaphorical cases but uses a decision boundary that admits too many false positives. Because proposal generation, verification, and adjudication are separate components, systematic calibration and stronger negative evidence may produce a more competitive balance without requiring an architectural redesign. This possibility must be tested on untouched evaluation data.

4.2 LIMITATIONS

The main language-support limitation concerns the verifier. The proposal stage can be used with any language supported by the configured LLM, but the verification stage requires language-specific lexical resources, dependency and POS analysis, masked-language-model probabilities, and calibrated thresholds. In the implemented detector, this verifier support is available only for English; other languages skip the verification evidence and are handled through proposal-only adjudication.

Another limitation concerns benchmark scope. The evaluation establishes the relative performance of `metaphoriq-detector` against the selected baselines on a bounded benchmark slice, but further validation is still necessary on larger corpora, additional genres, and historically distant texts. This is especially important for digital-humanities applications, where language use may differ substantially from the data distributions represented in common metaphor-detection benchmarks.

The most important empirical limitation is precision. The evaluated configuration's high-recall profile includes many false positives, especially when abstract but literal language resembles metaphorical usage. This indicates that the

verification and adjudication layers require stronger calibration, but the current experiment does not establish whether calibration alone is sufficient to reach the strongest baseline results.

4.3 FUTURE WORK

Future work should first expand the verification layer beyond English. This is not only a matter of adding more models. The detector would also need language-appropriate lexical resources, reliable POS and dependency parsers, masked-language models or equivalent probability estimators, and threshold calibration for each language. Multilingual support should therefore be treated as a new validation task, not as a direct configuration change.

A second direction is reducing false positives. The implemented detector uses fixed weights for combining SPV and MIP proxy scores and fixed thresholds for accepting metaphor, accepting literal, and forcing metaphor. A future version should tune these values on a development set and report sensitivity analyses, with precision as an explicit objective. This would require stronger negative evidence for literal abstract language, hard-negative examples that resemble metaphor without being metaphorical, and adjudication rules that can reject low-confidence proposals instead of treating most plausible figurative readings as positive cases. The disambiguation retry reduces unresolved outputs, but it does not by itself solve this precision problem.

Finally, the evaluation should be extended with additional datasets, expert annotation rounds, and historically distant texts. Larger evaluations would make it possible to analyze error types in more detail and to assess whether the structured metadata generated by the detector can support future supervised models or diagnostic annotation workflows.

5 BIBLIOGRAPHY

BIRKE, Julia; SARKAR, Anoop. A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language. *In:* Trento, Italy: Association for Computational Linguistics, 2006. Disponível em: <https://aclanthology.org/E06-1042/>

CHEN, Puli; YANG, Cheng; HUANG, Qingbao. Merely Judging Metaphor is Not Enough: Research on Reasonable Metaphor Detection. *In:* Association for Computational Linguistics, 2024. Disponível em: <https://aclanthology.org/2024.findings-emnlp.336/>

CHOI, Minjin *et al*. MelBERT: Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories. *In:* Online: Association for Computational Linguistics, 2021. Disponível em: <https://aclanthology.org/2021.naacl-main.141/>

CONNEAU, Alexis *et al*. Unsupervised Cross-lingual Representation Learning at Scale. *In:* Online: Association for Computational Linguistics, 2020. Disponível em: <https://aclanthology.org/2020.acl-main.747/>

CREATIVELANG. **Metaphor Detection RoBERTa Seq: token-level metaphor detection model**. , 2025.

DEVLIN, Jacob *et al*. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In:* Minneapolis, Minnesota: Association for Computational Linguistics, 2019. Disponível em: <https://aclanthology.org/N19-1423/>

HAYASHI, Takuya; SASAKI, Minoru. Applying Additional Auxiliary Context Using Large Language Model for Metaphor Detection. **Big Data and Cognitive Computing**, v. 9, n. 9, 2025.

HENGCHEN, Simon *et al*. NLP for Digital Humanities: Processing Chronological Text Corpora. *In:* Bangkok, Thailand: Association for Computational Linguistics, 2024. Disponível em: <https://aclanthology.org/2024.nlp4dh-1.10/>

HERRMANN, J. Berenike; REITER, Nils; PETRIS, Monica. Tool Criticism in Practice: On Methods, Tools and Aims of Computational Literary Studies. **Digital Humanities Quarterly**, v. 17, n. 2, 2023.

LAKOFF, George; JOHNSON, Mark. **Metaphors We Live By**. Chicago: University of Chicago Press, 1980.

LEONG, Chee Wee *et al*. A Report on the 2020 VUA and TOEFL Metaphor Detection Shared Task. *In:* Online: Association for Computational Linguistics, 2020. Disponível em: <https://aclanthology.org/2020.figlang-1.3/>

LI, Yucheng *et al*. FrameBERT: Conceptual Metaphor Detection with Frame Embedding Learning. *In:* Dubrovnik, Croatia: Association for Computational Linguistics, 2023. Disponível em: <https://aclanthology.org/2023.eacl-main.114/>

LIN, Yujie *et al*. A Dual-Perspective Metaphor Detection Framework Using Large Language Models. *In:* 2024. Disponível em: <https://arxiv.org/abs/2412.17332>

MILLER, George A. WordNet: A Lexical Database for English. **Communications of the ACM**, v. 38, n. 11, p. 39–41, 1995.

MOHAMMAD, Saif M.; SHUTOVA, Ekaterina; TURNEY, Peter D. Metaphor as a Medium for Emotion: An Empirical Study. *In:* Berlin, Germany: Association for Computational Linguistics, 2016. Disponível em: <https://aclanthology.org/S16-2003/>

THE PRAGGLEJAZ GROUP. MIP: A Method for Identifying Metaphorically Used Words in Discourse. **Journal of Pragmatics**, v. 39, n. 5, p. 563–576, 2007.

REIMANN, Sebastian; SCHEFFLER, Tatjana. The Struggles of Large Language Models with Zero- and Few-Shot (Extended) Metaphor Detection. **Journal for Language Technology and Computational Linguistics**, v. 38, n. 2, p. 97–109, 2025.

SANCHEZ-BAYONA, Elisa; AGERRI, Rodrigo. Metaphor and Large Language Models: When Surface Features Matter More than Deep Understanding. *In:* Vienna, Austria: Association for Computational Linguistics, 2025. Disponível em: <https://aclanthology.org/2025.findings-acl.898/>

SHUTOVA, Ekaterina; TEUFEL, Simone; KORHONEN, Anna. Metaphor Identification Using Selectional Preference Violation. *In:* Association for Computational Linguistics, 2013. Disponível em: <https://aclanthology.org/D13-1017/>

STECHLY, Kaya; VALMEEKAM, Karthik; KAMBHAMPATI, Subbarao. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks. *In:* 2025. Disponível em: <https://arxiv.org/abs/2410.05779>

STEEN, Gerard *et al*. **A Method for Linguistic Metaphor Identification: From MIP to MIPVU**. Amsterdam: John Benjamins, 2010.

TAHMASEBI, Nina; BORIN, Lars; JATOWT, Adam. A Survey of Computational Approaches to Linguistic Semantic Change. **Computational Linguistics**, v. 47, n. 3, p. 603–661, 2021.

UNIVERSIDADE FEDERAL DE SANTA CATARINA. IA E HISTÓRIA: GRUPO DE ESTUDOS E PESQUISA INTERDISCIPLINAR. **Informações institucionais**. , 2025.

YANG, Cheng *et al*. Rethinking Metaphor Evaluation: Aptness Judgments as a Cognitive Probe for Language Models. **Cognitive Computation**, v. 18, 2026.

APPENDIX A – SCIENTIFIC ARTICLE

A Dual-Stage Detection Pipeline for Metaphor Annotation Using LLM-Based Proposal Generation and Linguistic Verification

Gabriel Avila
Undergraduate Program in Information Systems
Federal University of Santa Catarina
Florianopolis, Brazil

Rodrigo Bragio Bonaldo
Department of History
Federal University of Santa Catarina
Florianopolis, Brazil

Jean Carlo Rossa Hauck
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianopolis, Brazil

***Abstract*—Metaphor annotation supports linguistics, discourse analysis, digital humanities, and natural language processing, but manual identification is difficult to scale and automatic predictions are often hard to audit. This paper presents `metaphoriq-detector`, a dual-stage pipeline that separates large language model proposal generation from deterministic linguistic verification. The proposal stage returns candidate spans and structured explanations based on the Metaphor Identification Procedure and semantic preference violation. The verification stage checks those proposals with part-of-speech and dependency analysis, a WordNet-derived proxy for basic-contextual meaning contrast, and masked-language-model anomaly scores. Fixed rules then adjudicate the evidence and trigger one disambiguation retry for uncertain cases. The detector was evaluated on 999 items drawn uniformly from VUA20, TroFi, and MOH-X and compared with XLM-R, CreativeLang RoBERTa, FrameBERT, and MelBERT. It obtained 0.9630 recall, 0.5148 precision, and 0.6710 F1. Although it did not surpass the strongest specialized baseline, the pipeline exposes candidate spans, explanations, verification metrics, and decision metadata. The results characterize a provisionally calibrated operating point and identify precision calibration as the principal remaining empirical problem.**

***Index Terms*—Metaphor detection, Large language models, Linguistic annotation, Natural language processing, Semantic preference violation**

I. Introduction

Metaphor annotation is relevant to linguistics, discourse analysis, digital humanities, and natural language processing (NLP), but it remains expensive to scale. Manual procedures such as the Metaphor Identification Procedure (MIP) and its MIPVU extension provide explicit criteria for deciding whether a lexical unit is metaphorical, yet they demand sustained work by trained annotators [1]; [2]. The problem becomes especially important in historical corpora, where meanings can change over time and annotations must remain interpretable for later expert review [3].

Neural models have improved automatic metaphor detection. Systems such as MelBERT combine contextual representations with metaphor-identification theory [4], while recent large language model (LLM) approaches can produce natural-language explanations and theory-guided judgments [5]; [6]. These capabilities do not eliminate the transparency problem. Direct model judgments may be influenced by surface cues, and asking a model to verify its own output does not provide independent evidence [7]; [8].

This paper presents `metaphoriq-detector`, a pipeline that treats the LLM output as a proposal rather than a final decision. An LLM first identifies candidate metaphorical spans and supplies structured MIP- and semantic-preference-based reasons. A separate deterministic verifier then computes lexical, syntactic, and semantic signals. Fixed adjudication rules combine the proposal and verifier confidence, with one retry for uncertain cases.

The work makes three contributions. First, it defines an architecture that separates generative interpretation from external verification. Second, it implements an inspectable output contract containing spans, explanations, verification scores, raw outputs, and decision metadata. Third, it evaluates the complete pipeline on uniform slices of VUA20, TroFi, and MOH-X against four metaphor-detection baselines. The resulting operating point has high recall but low precision. Because the configuration was selected through exploratory screening rather than systematic calibration, this result does not establish the performance ceiling of the architecture.

II. Background and Related Work

A. Linguistic Identification

MIP operationalizes metaphor identification through a contrast between basic and contextual meaning [1]. An annotator identifies a lexical unit, determines its meaning in context, checks whether it has a more basic meaning, and decides whether the contextual use can be understood by comparison with that basic sense. MIPVU expands the procedure with rules for lexical-unit boundaries, multiword expressions, and dictionary use [2]. This procedural view motivates two properties of the detector: decisions should preserve the relevant span, and the evidence should make the proposed basic-contextual contrast inspectable.

Semantic preference provides a complementary signal. Predicates tend to occur with arguments from particular semantic categories. A violation, such as an event normally associated with living entities being attributed to an abstract idea, can suggest a figurative interpretation. Selectional or

semantic preference violations have therefore been used in computational metaphor identification [9]. Such a violation is evidence rather than proof: unusual literal language can also be improbable, and conventional metaphors may no longer appear anomalous.

WordNet supplies reproducible lexical-sense information through synsets and glosses [10]. It cannot replace dictionary-based human analysis, but it can support an automatic proxy for the contrast between a default lexical sense and the word's surrounding context. Similarly, masked language models can estimate how expected a token is within a syntactic relation [11]. The proposed verifier combines these imperfect but external signals instead of asking the generating LLM to assess its own explanation.

B. Computational Detection

Contextual encoders made it possible to represent the same lexical form differently across usages. MelBERT explicitly models metaphor-identification perspectives through late interaction between representations [4]. FrameBERT incorporates frame information into metaphor detection [12], while multilingual and task-specific transformer encoders provide strong direct classification baselines [13]; [14].

Recent LLM work adds explicit reasoning and context. The dual-perspective framework of [5] prompts a model with MIP and semantic preference cues and includes model self-judgment. Explanation-oriented work asks models to provide reasons rather than only labels [6]. Auxiliary contextual sentences can also improve ambiguous metaphor decisions [15]. However, evaluations show that LLM metaphor judgments can depend strongly on lexical overlap, sentence length, and other surface features [7], and can overgeneralize in zero- and few-shot conditions [16].

The methodological gap addressed here is therefore not explanation generation alone. A free-form explanation and a self-judgment are still generated within the same model boundary. `metaphoriq-detector` retains the generative model's ability to formulate a hypothesis but evaluates that hypothesis using separately implemented linguistic signals.

III. Proposed Pipeline

A. System Overview

The detector receives a target sentence and optional neighboring sentences from the same document. It builds a small context window, requests a structured proposal from the LLM, sanitizes the returned spans, computes verifier evidence, and applies threshold-based adjudication. An uncertain decision triggers a second proposal with an explicit disambiguation instruction and, when available, a larger context window. The final output is always binary, but the internal record preserves the path used to reach it.

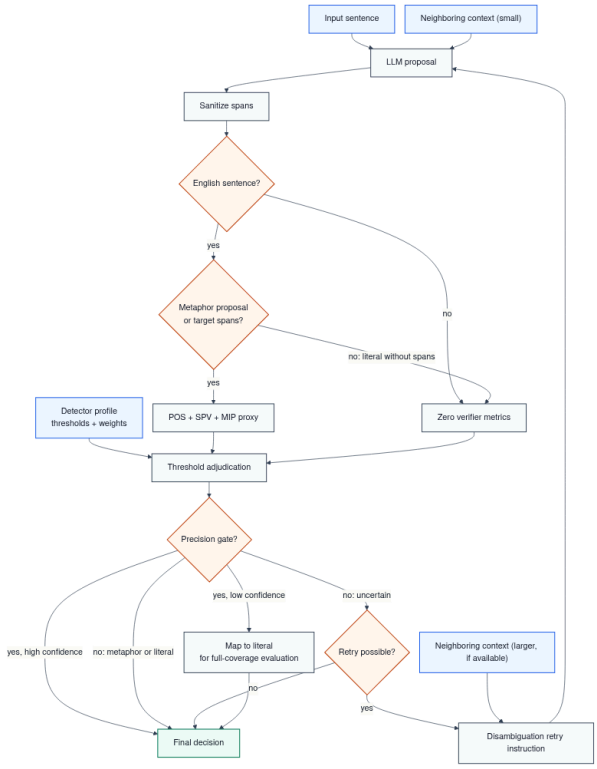


Fig. 1. Dual-stage metaphor detection workflow.

As shown in Figure 1, candidate generation and verification are distinct stages. This boundary also exists in the implementation: the detector package does not depend on stored benchmark reports, baseline environments, or historical experiment directories. Benchmark adapters invoke the detector, but evaluation artifacts do not influence its runtime decisions.

B. Proposal Generation

The proposal stage renders a prompt containing the target sentence, available context, and optional benchmark target information. In the evaluated experiments, the call used OpenAI's `gpt-5-nano` through the project's structured JSON generation service. The model is instructed to apply MIP and semantic preference cues and return strict JSON with three elements:

- a `metaphor` or `literal` label;
- spans copied from the target sentence, including text and character offsets;
- a reason object with MIP, semantic-preference, and literal-paraphrase fields.

Context is used only to disambiguate the target sentence. Returned spans may not come from neighboring text. The detector checks whether every offset matches the corresponding substring. If the text occurs in the sentence but the offsets are incorrect, it repairs them by exact search; otherwise, it drops the invalid span. This sanitization prevents generated locations from silently entering downstream annotation metadata.

C. External Verification

Verification is implemented for English sentences. Literal proposals without spans skip the expensive verifier and receive

zero-valued metrics. Non-English sentences use proposal-only adjudication because the available lexical and syntactic resources are English-specific.

For English proposals, the verifier first parses the sentence to obtain tokens, part-of-speech tags, and dependency relations. It then identifies mask candidates from structures including verb-subject and verb-object relations, copular complements, nouns in `of` constructions, and nouns with adjectival or compound modifiers. Each candidate is masked and scored by a masked language model. Low probability for the observed token produces a higher semantic-preference violation score. The detector retains the maximum anomaly score across the candidates.

The MIP proxy selects a content-word head for each proposed span, retrieves WordNet synsets, and compares the first synset gloss with the sentence context. Lower lexical overlap produces a larger proxy score. Selecting the first synset is a reproducible approximation, not a claim that it always corresponds to the basic meaning selected by a human annotator.

The verifier combines both signals as

$$C = 0.55S_{\text{SPV}} + 0.45S_{\text{MIP}},$$

where C is verifier confidence and both component scores are normalized. The verification record contains the component scores, combined confidence, enriched spans, and textual reason fields.

D. Adjudication and Retry

The evaluated configuration accepts an LLM metaphor proposal when verifier confidence is at least 0.4 and accepts a literal proposal when confidence is below 0.3. Confidence of at least 0.7 forces a metaphor decision even when the proposal is literal. All remaining cases become internally `uncertain` and enter the retry path.

The weights and thresholds are implemented design choices. They were selected through exploratory screening runs rather than through exhaustive search or statistical calibration. The screening stage evaluated 33 items from each of the three datasets. Candidate profiles varied target awareness, literal-target verification, thresholds, proposal-only behavior, MIP weighting, precision gating, and retry behavior. The retained profile represented a useful precision-recall operating point among those trials, but the evaluation below shows that it remains too permissive.

When source context exists, the first proposal uses one neighboring sentence on each side and the retry may use three on each side. The benchmark items used in this paper contain only individual sentences, so no additional document context was available. The retry still supplied an explicit instruction that the earlier adjudication was inconclusive and requested the strongest binary judgment. If threshold rules remain inconclusive after that pass, the retry proposal becomes the final label. Thus, uncertainty is diagnostic metadata rather than an exposed output class.

IV. Evaluation

A. Datasets and Baselines

The evaluation aggregates binary predictions from VUA20 [17], TroFi [18], and MOH-X [19]. Each dataset contributes 333 items, for 999 evaluated sentences in total. The equal-sized slices prevent the pooled metrics from being dominated by one dataset. The sample size was constrained by the cost and execution requirements of running the proposed detector and all baselines; it is not presented as a statistically optimized sample.

Although some source resources provide token- or span-level annotations, the shared task here is sentence-level binary detection: an item is metaphorical if the system identifies at least one metaphorical target usage. This aggregation permits comparison with XLM-R, CreativeLang RoBERTa, FrameBERT, and MelBERT under one prediction contract. The XLM-R and CreativeLang models run as Hugging Face token-classification pipelines. FrameBERT token outputs are grouped by sentence, and its borderline outputs remain non-binary. MelBERT target-token predictions are also aggregated to sentence identifiers.

All adapters normalize results into a shared representation containing a document identifier, model key, label, binary value, optional score, spans, and raw output. Gold items marked uncertain are excluded from binary metrics, as are model outputs that cannot be mapped to a binary label. The selected `metaphoriq-detector` profile returned a binary decision for every eligible row.

B. Results

TABLE I
Pooled binary benchmark metrics across VUA20, TroFi, and MOH-X.

Model	Prec.	Recall	F1	Acc.
Metaphoriq	0.5148	0.9630	0.6710	0.5395
XLM-R	0.5882	0.9589	0.7291	0.6527
CL RoBERTa	0.5905	0.9651	0.7327	0.6567
FrameBERT	0.5877	0.8481	0.6943	0.5903
MelBERT	0.6798	0.8850	0.7690	0.7407

Table I shows that MelBERT obtains the strongest pooled F1 at 0.7690, with precision 0.6798 and recall 0.8850. CreativeLang RoBERTa and XLM-R follow with F1 scores of 0.7327 and 0.7291. `metaphoriq-detector` reaches F1 0.6710, below those three baselines and below FrameBERT's 0.6943. FrameBERT's coverage is only 0.4324 because its borderline outputs are not forced into binary labels, so its pooled result is not directly equivalent to a full-coverage operating point.

The defining behavior of the proposed detector is recall 0.9630 combined with precision 0.5148. It produces 469 true positives and only 18 false negatives, but also 442 false positives and only 70 true negatives. It therefore recovers most metaphorical items while frequently assigning metaphorical readings to literal abstract language.

The dataset-level F1 scores further qualify the aggregate result. The detector obtains 0.9333 on VUA20, 0.5890 on

TroFi, and 0.3596 on MOH-X. This variation indicates that one set of fixed proxy weights and thresholds does not transfer uniformly across the three benchmark distributions.

V. Discussion

The evaluation supports a narrower claim than full automatic superiority. The pipeline did not outperform MelBERT, CreativeLang RoBERTa, or XLM-R by pooled F1, and FrameBERT obtained a numerically higher F1 under partial coverage. The result therefore characterizes the evaluated configuration rather than the performance ceiling of the architecture.

The observed imbalance should not be treated as an intrinsic recall-oriented property of the detector. The confidence weights and adjudication thresholds were selected through exploratory experiments on a small screening sample, without systematic optimization on a separate development set. The current evaluation does not show whether the 442 false positives result primarily from weak verifier signals or from an inadequate decision boundary. It establishes that the present configuration is insufficiently calibrated, but not that the architecture cannot achieve a more competitive operating point.

This distinction matters because proposal generation, verification, and adjudication are independent components. The decision boundary and verifier evidence can be revised without replacing the proposal mechanism. Related systems show that metaphor-theoretic information can support competitive detection when it is learned or appropriately calibrated [4]; [12]; [5]. The next evaluation should therefore calibrate weights and thresholds on a disjoint development set, include stronger negative evidence for abstract literal language, and report sensitivity or precision-recall analyses. If these changes reduce false positives while preserving most of the current recall, the pipeline may approach or surpass the selected baselines. Establishing that result requires a new evaluation on untouched data.

Several limitations constrain the findings. First, verification is English-only; non-English inputs fall back to proposal-only decisions. Second, the WordNet score uses the first synset and gloss overlap as a proxy rather than implementing full human MIP analysis. Third, the semantic-preference score depends on masked token probabilities over a selected set of dependency patterns. Both signals are reproducible but incomplete. Fourth, thresholds were selected through small pilot runs and were not calibrated on a dedicated development set. Finally, the evaluation uses bounded, sentence-only slices of three conventional datasets and does not establish performance on historical documents, other genres, or larger corpora.

The immediate technical priority is precision calibration. Future experiments should tune weights and thresholds on a separate development set, report sensitivity curves, and include hard negative examples of abstract but literal language. Learned calibration could be compared with the current rules while retaining the external evidence boundary. Verification should also be expanded with language-specific lexical resources, parsers, and probability estimators, rather than assuming that the English configuration transfers unchanged.

VI. Conclusion

This paper presented a dual-stage metaphor detection pipeline that combines LLM proposal generation with deterministic linguistic verification. The design preserves the interpretive capability of a generative model while exposing candidate spans, structured reasons, WordNet- and semantic-preference-based signals, and adjudication metadata. On 999 benchmark items, the detector reached 0.9630 recall, 0.5148 precision, and 0.6710 F1. It did not surpass the strongest specialized baselines and produced too many false positives for unreviewed binary classification. The results instead characterize a provisionally calibrated operating point: the modular design makes stronger calibration and verifier revision plausible without architectural replacement, but that possibility must be tested on untouched evaluation data.

The implementation, benchmark adapters, experiment scripts, and stored artifacts are available at https://codigos.ufsc.br/avila.g/metaphoriq-detector.

References

[1] The Pragglejaz Group, "MIP: A Method for Identifying Metaphorically Used Words in Discourse," *Journal of Pragmatics*, vol. 39, no. 5, pp. 563–576, 2007, doi: 10.1016/j.pragma.2006.11.001.

[2] G. Steen, A. Dorst, B. Herrmann, A. Kaal, T. Krennmayr, and T. Pasma, *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*. Amsterdam: John Benjamins, 2010.

[3] N. Tahmasebi, L. Borin, and A. Jatowt, "A Survey of Computational Approaches to Linguistic Semantic Change," *Computational Linguistics*, vol. 47, no. 3, pp. 603–661, 2021, doi: 10.1162/coli_a_00413.

[4] M. Choi *et al.*, "MelBERT: Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, 2021, pp. 1763–1773. doi: 10.18653/v1/2021.naacl-main.141.

[5] Y. Lin, J. Liu, Y. Gao, A. Wang, and J. Su, "A Dual-Perspective Metaphor Detection Framework Using Large Language Models," in *Proceedings of the 2024 Conference* (*arXiv preprint*), 2024. [Online]. Available: https://arxiv.org/abs/2412.17332

[6] P. Chen, C. Yang, and Q. Huang, "Merely Judging Metaphor is Not Enough: Research on Reasonable Metaphor Detection," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Association for Computational Linguistics, 2024, pp. 5850–5860. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.336/

[7] E. Sanchez-Bayona and R. Agerri, "Metaphor and Large Language Models: When Surface Features Matter More than Deep Understanding," in *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria: Association for Computational Linguistics, 2025, pp. 17462–17477. doi: 10.18653/v1/2025.findings-acl.898.

[8] K. Stechly, K. Valmeekam, and S. Kambhampati, "On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks," in *International Conference on Learning Representations*, 2025. [Online]. Available: https://arxiv.org/abs/2410.05779

[9] E. Shutova, S. Teufel, and A. Korhonen, "Metaphor Identification Using Selectional Preference Violation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2013, pp. 171–182. [Online]. Available: https://aclanthology.org/D13-1017/

[10] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995, doi: 10.1145/219717.219748.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[12] Y. Li, S. Wang, C. Lin, F. Guerin, and L. Barrault, "FrameBERT: Conceptual Metaphor Detection with Frame Embedding Learning," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia: Association for Computational Linguistics, 2023, pp. 1558–1563. doi: 10.18653/v1/2023.eacl-main.114.

[13] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 8440–8451. doi: 10.18653/v1/2020.acl-main.747.

[14] CreativeLang, "Metaphor Detection RoBERTa Seq: token-level metaphor detection model." 2025.

[15] T. Hayashi and M. Sasaki, "Applying Additional Auxiliary Context Using Large Language Model for Metaphor Detection," *Big Data and Cognitive Computing*, vol. 9, no. 9, 2025, doi: 10.3390/bdcc9090218.

[16] S. Reimann and T. Scheffler, "The Struggles of Large Language Models with Zero- and Few-Shot (Extended) Metaphor Detection," *Journal for Language Technology and Computational Linguistics*, vol. 38, no. 2, pp. 97–109, 2025, doi: 10.21248/jlcl.38.2025.287.

[17] C. W. Leong, B. Beigman Klebanov, C. Hamill, E. W. Stemle, R. Ubale, and X. Chen, "A Report on the 2020 VUA and TOEFL Metaphor Detection Shared Task," in *Proceedings of the Second Workshop on Figurative Language Processing*, Online: Association for Computational Linguistics, 2020, pp. 18–29. doi: 10.18653/v1/2020.figlang-1.3.

[18] J. Birke and A. Sarkar, "A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language," in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy: Association for Computational Linguistics, 2006. [Online]. Available: https://aclanthology.org/E06-1042/

[19] S. M. Mohammad, E. Shutova, and P. D. Turney, "Metaphor as a Medium for Emotion: An Empirical Study," in *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, Berlin, Germany: Association for Computational Linguistics, 2016, pp. 23–33. doi: 10.18653/v1/S16-2003.