Loci Similes: A Benchmark for Extracting Intertextualities in Latin Literature*

Anonymous ACL submission

Abstract

Tracing connections between historical texts is an important part of intertextual research, enabling scholars to reconstruct the *virtual library* of a writer and identify the sources influencing their creative process. These intertextual links manifest in diverse forms, ranging from direct verbatim quotations to subtle allusions and paraphrases disguised by morphological variation. Language models offer a promising path forward due to their capability of capturing semantic similarity beyond lexical overlap. However, the development of new methods for this task is held back by the scarcity of standardized benchmarks and easy-to-use datasets. We address this gap by introducing *Loci Similes*, a benchmark for Latin intertextuality detection comprising a curated dataset of ∼176k text segments and 1,490 expert-verified parallels, including 945 labeled references from an existing dataset. Using this data, we establish baselines for retrieval and classification of intertextualities with pretrained encoder language models.

1 Introduction

Identifying intertextual connections between documents is an important task in classical philology, as it reveals how later works engage with earlier texts and traditions. For centuries, scholars detected intertextual references by relying on memory and the manual collation of *Loci Similes*, i.e., parallel passages that exhibit lexical, semantic, or thematic resemblance. Although digitization has augmented this process through lexical search tools, most approaches still depend on exact n-gram matching or heuristic filtering (Schropp et al., 2024a). This limits discovery rates in ancient texts, where intertextuality typically manifests itself not as verbal quotation, but as subtle allusion, paraphrase, or thematic variation (Manjavacas et al., 2019; Gong et al., 2025), often complicated by orthographic volatility (Miller et al., 2025).

Example of Historical Text Reuse

Source: Virgil, *Aeneid* 2.774
Context: Aeneas is terrified when the ghost of his wife Creusa appears to him during the burning of Troy.

> "... obstipui, steteruntque comae et **uox faucibus haesit**."
> *(... I was stupefied, my hair stood on end, and my **voice stuck in my throat**.)*

Reuse: Jerome, *Epistula* 130.5.5
Context: A family's shocked reaction to Demetrias's vow of Christian virginity.

> "**Haesit uox faucibus** et inter ruborem atque pallorem metumque ac laetitiam cogitationes uariae mutabantur."
> *(**The voice stuck in their throat**, and between blushing and pallor, fear and joy manifold thoughts kept shifting.)*

Figure 1: **Example of intertextual reference.** Reuse of a classic Virgilian phrase for speechlessness by Jerome. While retaining the semantic core, the author alters the word order to adapt the expression to a different context.

Recovering such textual reuses is not merely a matter of identifying sources. It facilitates research on broader cultural-historical phenomena (Tangherlini and Chen, 2024). In particular, it supports work on reception and cultural hybridization in Late Antiquity, where pagan texts persist as the rhetorical substrate of elite writing while being recontextualized within emerging Christian discourse. Classical forms often remain recognizable even as their functions shift toward Christian meaning-making, a phenomenon visible in both syntactic stylometry (Gorman and Gorman, 2016; Bernardo et al., 2021) and semantic motifs. In this sense, quotation patterns suggest how Christian authors do not abandon classical texts and their cultural contexts, but reuse their language, redirecting its meanings and connotations within Christian interpretive frameworks.

*Code and data are available at https://anonymous.4open.science/r/locisimiles-2338 (anonymized for review).

A case in point is the Church Father Jerome. Recent digital-hermeneutic studies have begun to map his "micro-quotations" (Schropp et al., 2024a), yet the semantic breadth of his reuse remains a challenge. When Jerome alludes to the Augustan poet Virgil, he often retains the semantic core of a hexameter verse while altering its word order or syntax to suit his Christian prose context. In Jerome's writings, the general tension between his pagan *paideia* and Christian discourse emerges with particular clarity in the details of how he quotes classical pagan sources and adapts them for his own texts (see Figure 1). Since canonical authors such as Virgil were deeply embedded in the educational curriculum, quoting them often served as a shorthand for shared cultural memory. Late Antique Christian writers like Jerome inherit this repertoire, but their reuse frequently reframes pagan language within Christian contexts, making citation patterns a measurable trace of shifting cultural authority.

Beyond extraction, systematically mapping these dependencies allows scholars to reconstruct the "virtual library" available to an author and gaining insight into which sources most strongly shaped their writing. Analyzing how these connections cluster, from explicit citation to subtle echo, further helps refine theoretical definitions of intertextuality and probe the rhetorical motivations behind text reuse. Although some resources exist (Burns et al., 2021; Toyin et al., 2026), automated, language-model-based detection of intertextual connections remains constrained by a lack of standardized benchmarks and accessible datasets.

In this paper, we take a step towards addressing this gap by introducing *Loci Similes*, a benchmark designed to enable researchers to systematically compare and evaluate computational approaches for intertextuality detection. Our paper makes the following key contributions:

- **Curated benchmark dataset** of $\sim$176k Latin text segments, partitioned into a query and source corpora, accompanied by a ground truth dataset of $1,490$ expert-verified intertextual links.
- **Evaluation framework** that aligns more closely with the practical constraints of philological workflows.
- **Baseline results** for retrieval models, classification models, and a retrieve-and-rerank pipeline for intertextuality detection, serving as a foundation for future comparisons.

2 Related Work

2.1 Intertextuality Detection in Latin

Work on Latin intertextuality spans a spectrum of methods, from surface-level matching to increasingly semantic representations. Revellio (2022) developed a rule-based approach operating directly on surface strings to uncover Virgilian references in Jerome's letters, and Schropp et al. (2024a) refined this with n-gram matching pipelines and cascaded filters to detect very short references. Such methods capture verbatim and near-verbatim reuse well, but miss semantically related passages that share little surface vocabulary.

Word-level embeddings instead capture semantic similarity beyond lexical overlap. Burns et al. (2021) used static Word2Vec models trained on lemmatized text to rank intertextual phrases, evaluated against 945 parallels from Valerius Flaccus' *Argonautica* (Dexter et al., 2024). Manjavacas et al. (2019) approached allusion detection in Latin sermons as information retrieval using fastText embeddings, and Manjavacas et al. (2020) extended this work on the *Patrologia Latina*, modeling lexical and thematic similarity as separate axes.

Gong et al. (2025) replaced static vectors with contextual LatinBERT (Bamman and Burns, 2020) embeddings, comparing source words to target bigrams to identify allusions in Lucan's *Pharsalia*. Operating at the sentence level, Riemenschneider and Frank (2023a) applied Sentence Transformers (Reimers and Gurevych, 2019) to ancient languages, introducing SPhilBERTa, a multilingual model fine-tuned on parallel sentences in Ancient Greek, Latin, and English (Riemenschneider and Frank, 2023b). More recently, Toyin et al. (2026) introduced Gretino, a benchmark for semantic retrieval in Latin, Ancient Greek, and cross-lingual settings. Unlike these resources, *Loci Similes* labels positive links by reference type, allowing verbatim references and allusions to be evaluated separately.

2.2 Text Reuse in Other Languages

Beyond Latin, text-reuse detection spans many languages and tasks. Some frameworks pair graph-based modeling (Kuznetsov et al., 2022) with large language model (LLM)-assisted extraction (Lau and McManus, 2024), and contextual models can outperform lexical baselines (MacLaughlin et al., 2021). For English, Rosson et al. (2023) built a tool for exploring reuse in Early Modern British texts, and Mahadevan et al. (2025) scaled detection

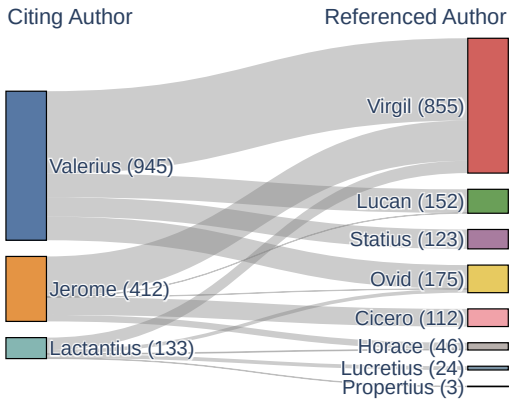


Figure 2: **Distribution of confirmed references.** Manually verified intertextual links in the annotated dataset by citing author (Jerome, Lactantius, and Valerius Flaccus) and source author (including Virgil, Cicero, and others).

to billions of instances.

Specialized approaches address low-resource languages. Miller et al. (2025) enhanced reuse detection in Hebrew and Aramaic manuscripts using fastText embeddings, and Gorman and Gorman (2016) modeled the authorial style in Ancient Greek through unsupervised clustering. Sharjeel et al. (2023) addressed cross-lingual reuse in English–Urdu using machine translation.

2.3 Further Related Tasks

Intertextuality detection shares concepts with quote detection (Janicki et al., 2023) and paraphrase identification (Vrbanec and Mestrovic, 2023). However, quote detection targets explicitly marked citations (Petersen-Frey and Biemann, 2024; Zhang and Liu, 2022), and paraphrase detection focuses on semantic equivalence (Wahle et al., 2022). Our task differs in requiring the capture of unmarked allusions and recontextualized echoes at the segment level, across otherwise unrelated documents.

3 Dataset Construction

While traditional scholarship has documented numerous Latin intertextual parallels, computational research remains constrained by the lack of standardized benchmarks. Aggregating these references is non-trivial, as the data is dispersed across commentaries, *indices locorum* ("back-of-the-book indexes"), and philological case studies. To address this gap, we curated a corpus of ∼176k text segments spanning works by multiple Latin authors (see Table 1) and a merged ground truth dataset of 1,490 confirmed intertextual links (see Figure 2).

Author	Segments	Avg. Tokens	Min	Max	Std. Dev.
Query Corpus					
Jerome	74,672	31.13	1	617	22.32
Lactantius	8,444	28.53	1	369	19.48
Valerius	2,138	29.60	1	145	18.10
Total / Avg.	***85,254***	***30.83***	***1***	***617***	***21.97***
Source Corpus					
Cicero	54,331	28.97	1	1,377	25.40
Ovid	14,096	26.51	1	432	19.53
Virgil	4,861	29.65	2	353	19.55
Martial	4,114	26.13	2	274	21.07
Lucan	2,952	31.26	2	245	21.83
Statius	2,527	44.08	2	412	40.22
Horace	2,353	33.60	2	195	24.83
Propertius	1,889	22.18	2	267	17.56
Lucretius	1,826	40.95	3	256	27.40
Catullus	809	27.43	2	149	22.08
Tibullus	788	26.12	2	251	20.70
Total / Avg.	***90,546***	***29.17***	***1***	***1,377***	***24.61***

Table 1: **Corpus statistics by author.** Breakdown of the dataset into *Query* (citing authors) and *Source* (referenced authors) corpora, showing the number of segments and token statistics for each author.

3.1 Latin Corpus Curation

We compose the corpus of a separate *Query Corpus* and a *Source Corpus*. The former comprises works by the Late Antique authors Jerome and Lactantius, together with Book 1 of Valerius Flaccus' *Argonautica* added through the subset from Burns et al. (2021), totaling ∼85k text segments. The latter, which serves as the retrieval target, consists of ∼91k segments drawn from eleven canonical classical Latin authors: Cicero, Lucretius, Catullus, Virgil, Horace, Tibullus, Propertius, Ovid, Lucan, Martial, and Statius. A detailed breakdown of segment counts and token statistics for each author is presented in Table 1. The texts were aggregated from three primary digital repositories: *Corpus Corporum*, the *Tesserae Project*, and the *OpenGreekandLatin Project*. For a complete listing of the specific works, critical editions utilized, and their respective provenance, see Appendix A. Since we exclusively rely on publicly available texts to compile the corpus, we release the corpus files alongside the ground-truth dataset.

3.2 Ground Truth Construction

To construct our merged ground-truth dataset, we combine 545 links that we curated for *Loci Similes* with an existing dataset of 945 labeled references from Valerius Flaccus' *Argonautica* used by Burns et al. (2021), yielding 1,490 links in total.

First, we sourced 270 references to Virgil and

Taxonomy of Intertextual Links

1. Verbatim Reference (*cit.*) – Literal Reuse

Near-verbatim reuse sharing at least two congruent lemmata in a coherent context.

- *Marked:* Explicitly attributed to a specific author or source (e.g. *ut ait Cicero...* → "as Cicero says").
- *Unmarked:* Silent formatting or incorporation of exact text strings from other sources.
- *Paraphrased:* Morphological, word-order, or lemma-equivalent variation (*arma uirumque* → *armis uirisque*; *cognoscat* → *cognoscere*).

2. Allusion (*cf.*) – Semantic Similarity

Thematic or stylistic echo without close lexical overlap; synonyms, paraphrase, or shared imagery instead of shared lemmata.

- *Single reference:* Shared distinct vocabulary or imagery recontextualized in a new setting (*infans* → *puer*; *paruulus* → *paruus*).
- *Systemic:* Broad stylistic, rhythmic, or thematic imitation on multiple occasions in the document.

Figure 3: **Annotation scheme.** Following Schropp et al. (2024b), we adopt a two-way reference-type scheme based on *Verbatim References* (*cit.*) and *Allusions* (*cf.*).

Cicero in Jerome's *Epistulae* from the dataset[1] established by Schropp et al. (2024b), adapting these entries to sentence-level granularity and excluding links disputed by our expert annotators. We added 275 further links, identified with the rule-based *n-gram* matching approach of Schropp et al. (2024a). Their pipeline scans for shared, non-contiguous tokens within a fixed window, then refines the candidate pairs through a cascade of filters that remove stopwords, enforce part-of-speech constraints, and exclude generic collocations by embedding similarity. Domain experts then manually evaluated the candidates, retaining only those confirmed as genuine intertextual links.

Second, we incorporated the dataset of Burns et al. (2021), comprising 945 known references between Book 1 of the *Argonautica* and four major Latin epics: Virgil's *Aeneid*, Ovid's *Metamorphoses*, Lucan's *Pharsalia*, and Statius' *Thebaid*. These references derive from traditional scholarship, namely the commentaries of Spaltenstein (2002), Kleywegt (2005), and Zissos (2008). We aligned their granularity and format with our existing dataset.

3.3 Annotation Process

Although the pipeline's filtering removed many false positives, numerous candidates still involved common collocations (e.g., *puncto temporis* "in a moment") and required manual exclusion. Therefore, four experts in Latin literature (two pre-PhD, two post-PhD) annotated the candidates, assessing whether each overlap was a meaningful intertextual reference or merely coincidental. For every pair confirmed as positive, annotators then assigned a reference type, distinguishing verbatim references from allusions. Together with unrelated pairs, this second stage yields a multiclass labeling into *no match*, *verbatim reference*, and *allusion*, enabling finer evaluation, since explicit lexical reuse, looser allusive correspondences, and pairs labeled *no match* pose substantially different retrieval and classification problems.

We applied the same scheme across all sources: reusing the reference-type labels for the 270 references from Schropp et al. (2024b) and manually annotating the 275 *Loci Similes* links and the 945 from Burns et al. (2021). Figure 3 describes the two positive reference types. Annotation conflicts were resolved through group discussion. The first stage was guided by three criteria:

1. **Use of Uncommon Vocabulary:** If the lexical overlap consists of rare or marked words, the likelihood increases that these elements were deliberately borrowed.

2. **Attested Frequency:** A specialized Latin corpus database[2] served to help determine the frequency of overlapping expression in Latin literature. Expressions appearing exclusively in the candidate source and target passages were treated as strong indicators of a unique intertextual relationship.

3. **Conduit Function:** Most importantly, if the cited expression contributes semantic, rhetorical, or cultural information from the source text that cannot be derived from the target passage in isolation, thereby implicitly enriching the interpretation, it is viewed as intertextual.

We release the complete dataset, including the query texts, source corpus, and annotated intertextual links, as a Hugging Face dataset.[3]

[1] https://doi.org/10.11588/data/FVCULR

[2] http://clt.brepolis.net/llta/Search

[3] https://anonymous.4open.science/r/locisimiles-2338. Link to code and data is anonymized for review; the final release will be hosted on Hugging Face.

Model	Verbatim (cit.) Rec.			Allusion (cf.) Rec.		
	10	100	1000	10	100	1000
TF-IDF (surf. 1g)	0.61	0.75	0.88	0.09	0.24	0.51
TF-IDF (lem. 1g)	0.55	0.72	0.90	0.08	0.21	0.57
TF-IDF (lem. 1+2g)	0.58	0.74	**0.91**	0.08	0.26	**0.61**
BM25 (surf. 1g)	**0.68**	0.76	0.87	**0.12**	0.28	0.50
BM25 (lem. 1g)	0.64	**0.78**	0.90	**0.12**	**0.29**	0.58
BGE-M3	0.50	0.64	0.78	0.08	0.21	0.44
E5-large	0.50	0.63	0.80	0.07	0.18	0.48
E5-base	0.49	0.61	0.77	0.06	0.16	0.44
E5-small	0.44	0.58	0.75	0.06	0.14	0.35
Granite-278m	0.43	0.54	0.69	0.03	0.12	0.36
Granite-107m	0.31	0.41	0.58	0.02	0.07	0.27
SPhilBERTa	0.41	0.56	0.74	0.06	0.19	0.48
Word2Vec (WE, 2g)	0.15	0.31	0.61	0.01	0.07	0.25
LatinBERT (WE)	0.13	0.17	0.25	0.01	0.02	0.06

Table 2: **Performance comparison of retrieval approaches.** Lexical approaches (TF-IDF, BM25) use surface tokens (*surf*) or lemmas (*lem*) with unigrams (*1g*) or unigrams+bigrams (*1+2g*); WE denotes approaches based on word embeddings.

Model	Verbatim (cit.)			Allusion (cf.)		
	P	R	F1	P	R	F1
LogReg (lexical)	0.37	0.83	0.46	0.02	0.37	0.04
GBDT (lexical)	0.36	**0.84**	0.45	0.02	0.22	0.04
mmBERT Base	0.75	0.75	0.74	0.09	0.45	0.13
mmBERT Small	0.72	0.77	0.72	**0.09**	0.37	**0.13**
XLM-R Large	**0.75**	0.81	**0.76**	0.07	0.53	0.11
XLM-R Base	0.69	0.75	0.69	0.07	**0.56**	0.11
mBERT Large	0.71	0.78	0.72	0.08	0.36	0.12
mBERT Base	0.69	0.75	0.69	0.08	0.42	0.12
BERT-Romanian	0.62	0.71	0.63	0.07	0.42	0.11
PhilBerta	0.62	0.73	0.64	0.03	0.45	0.05
LaBerta	0.25	0.66	0.32	0.02	0.24	0.03
LatinBERT	0.06	0.57	0.10	0.02	0.17	0.03
RoBERTa-Latin	0.01	0.26	0.02	0.01	0.12	0.02

Table 3: **Performance comparison of classification approaches.** Lexical approaches use sparse features; the remaining rows are fine-tuned transformer encoders. Decision thresholds are tuned per class and model the training set.

4 Experimental Setup

We use the dataset of Section 3 to answer three questions: (i) how well do retrieval-only methods rank true source segments among many candidates; (ii) how well do classifiers separate verbatim references, allusions, and pairs labeled *no match* on a fixed candidate set; and (iii) what precision–efficiency trade-off can a retrieve-and-rerank pipeline reach.

Task and pipeline. We define intertextuality detection as the identification of directional dependencies between a *query set* (segments from chronologically later works) and a *source set* (segments from earlier works). Each query segment maps to zero, one, or multiple source segments, either as a verbatim reference or as an allusion. Comparing 85k query against 91k source segments yields roughly 7.7 billion pairs, so exhaustive multiclass classification of every pair is computationally intractable. We therefore evaluate three method families: a retrieval-only approach that ranks candidates by similarity, a classification-only approach applied to a fixed candidate set, and a retrieve-and-rerank pipeline that classifies the top-k candidates produced by a retriever as *no match*, *verbatim reference*, or *allusion*.

4.1 Retrieval Models

We compare three retriever families. The sparse lexical methods TF-IDF and BM25 rank source segments by similarity over surface tokens or Classical Language Toolkit (CLTK) lemmas, with unigrams or unigrams+bigrams, and use no learned parameters. Two word-embedding (WE) baselines reimplement prior work at the word level: following Burns et al. (2021), the mean cosine similarity of bigram Word2Vec vectors, and following Gong et al. (2025), position-aggregated LatinBERT token embeddings. The dense retrievers are bi-encoders fine-tuned with Online Contrastive Loss (Hadsell et al., 2006); we evaluate multilingual encoders (BGE-M3, E5 small/base/large, Granite-107m/278m) and the Latin-specific SPhilBERTa. Following Su et al. (2023), query and candidate are encoded with distinct `Query:` and `Candidate:` prefixes. Grouped model descriptions and training details are in Appendix D.

4.2 Classification Models

For classification we compare two families. As lexical reference points we train a logistic regression (LogReg) and a histogram gradient-boosted decision tree (GBDT) on per-pair similarity features: TF-IDF cosine over lemma unigrams, lemma unigrams+bigrams, and character 3–4-grams, together with Jaccard overlap, raw-token overlap, and length features, all computed with the same CLTK pipeline as the sparse retrievers. Our main classifier is a cross-encoder that processes the query–candidate pair as a single concatenated input (Reimers and Gurevych, 2019), so self-attention can compare the two segments at the token level, which matters most for distinguishing allusions from verbatim references. We evaluate multilingual cross-encoders, namely mmBERT,

a modern multilingual encoder, XLM-RoBERTa (XLM-R), ModernBERT (mBERT), and BERT-Romanian, together with Latin-specific cross-encoders (PhilBerta, SPhilBERTa, LaBerta, LatinBERT, RoBERTa-Latin). Table 3 reports the small/base or base/large variants where applicable. All cross-encoders are fine-tuned for 4 epochs with batch size 32 and learning rate 2×10^{-5}; full feature definitions and training details are in Appendix E. The classifier is three-class throughout (*no match*, *verbatim reference*, *allusion*); for retrieval-style evaluation (Recall@k and overall reference-detection F1) the two positive classes are merged into a single positive label, while ref-type-specific metrics are obtained by filtering the same predictions.

4.3 Metrics

Because most query–source pairs are non-references, accuracy is uninformative: a trivial all-negative predictor already scores near-perfectly. We therefore report per-class precision (P), recall (R), and F1 for verbatim references and allusions. In addition, we report three error-based metrics normalized by the total number of pairs (N): the Segment-Misclassification Rate ($\mathrm{SMR} = (\mathrm{FP}+\mathrm{FN})/N$) as a global error rate, the Global False-Positive Rate ($\mathrm{FPR} = \mathrm{FP}/N$) for spurious matches, and the Global False-Negative Rate ($\mathrm{FNR} = \mathrm{FN}/N$) for missed references. For retrieval we additionally report Recall@k at $k \in \{10, 100, 1000\}$. Formal definitions are given in Appendix B.

4.4 Data Splits

We use 5-fold cross-validation on the 1,490 verified positive pairs, with all targets of a given query assigned to the same fold; fold sizes therefore vary slightly. At evaluation we reconstruct a realistic retrieval scenario: each fold contains a query document (citing query segments plus an equal number of in-domain distractors, doubling the total) and a source document (cited source segments plus an equal number of non-cited segments drawn from the same source works, again doubling the total), and every query segment is scored against every source segment in the fold. Per-fold sizes are reported in Table 4: on average each fold spans ∼282 queries and ∼565 sources, yielding ∼159,000 candidate pairs of which ∼298 are ground-truth positives (~0.19%) and the remainder serve as in-domain negatives.

	Queries		Sources		Pairs	
Fold	**Citing**	**Total**	**Cited**	**Total**	**Pos.**	**Neg.**
0	142	284	269	538	279	152,513
1	141	282	254	508	275	142,981
2	141	282	319	638	336	179,580
3	141	282	301	602	317	169,447
4	141	282	269	538	283	151,433
Total	706	1,412	1,412	2,824	1,490	795,954
Mean	141.2	282.4	282.4	564.8	298.0	159,191

Table 4: **Per-fold sizes of the evaluation files.** *Citing/cited* segments participate in at least one ground-truth match; an equal number of in-domain distractors is added 1:1, so the *Total* is always twice the citing/cited count. Under *Pairs*, *Pos.* counts annotated matches and *Neg.* the remaining scored pairs.

4.5 Model Configuration and Ablation

Each training fold is augmented with n negative pairs at positive-to-negative ratios from 1:1 to 1:10. For the dense retriever and the cross-encoder classifier we sweep base model family, negative-sampling ratio, and learning rate / epochs over the same 5 folds; sparse and WE methods need no fine-tuning and use default configurations. Full grids are in Appendices D and E, with the corresponding sweep results in Appendices G and H. Evaluation code and pipelines are released as the open-source Python package `locisimiles` (Appendix K).

5 Experimental Results

In this section, we present our quantitative findings.

5.1 Information Retrieval Results

Lexical methods match or outperform dense neural retrievers (Table 2): BM25 with lemmatized unigrams achieves the best Recall@100 on both reference types. Among neural models, large multilingual encoders clearly outperform native Latin baselines, with BGE-M3 leading the dense retrievers. Allusion recall stays far below verbatim throughout. Full results and ablations on negative sampling and hyperparameters are in Appendix G.

5.2 Classification Results

Transformer encoders clearly outperform the lexical baselines (Table 3). LogReg and GBDT achieve high verbatim recall but low precision, and near-zero precision on allusions. XLM-R Large performs best on verbatim references, mmBERT on allusions. Allusions remain far harder than verbatim references throughout, limited by low precision

despite moderate recall. Full results and ablations on negative sampling and hyperparameters are in Appendix H.

5.3 Retrieve-and-Rerank Results

Guided by the per-stage findings, we instantiate the retrieve-and-rerank pipeline of Section 4 with BM25 over lemma unigrams as the first stage, the strongest retriever on Recall@100 in Section 5.1, and the cross-encoder classifier as the reranker, which Section 5.2 showed dominates the lexical baselines on precision. As a reference point we compare against a retrieval-only baseline that treats all top-k candidates as detected references.

The pipeline sharply reduces false positives relative to retrieval-only (Figure 4): at $k = 5$ it reaches F1 0.45 versus 0.26, and holds F1 0.45 at $k = 100$, where retrieval-only collapses to 0.04. In practical terms, at $k = 100$ the pipeline recovers 63% of true intertextual references (185 out of 294) while narrowing the candidate set from 56,480 to 1,901 passages, a 97% reduction in the space a philologist would need to inspect (Appendix I).

6 Error Analysis

To complement the aggregate results in Section 5, we analyze where XLM-R Large succeeds and fails. We break performance down by author, reference type, and lexical overlap, then inspect a representative false positive and false negative in detail.

6.1 Quantitative Breakdown

The per-author-pair breakdown (Table 5) shows that classification quality is governed largely by lexical overlap. For verbatim references the relationship is roughly monotonic: detection of high-overlap pairs is essentially a solved task, while the low-overlap Valerius pairs are the weakest. The same factor explains the verbatim–allusion gap, since allusions share far fewer tokens. Within the allusion class, however, overlap no longer predicts performance: Valerius allusions outperform Jerome's despite lower overlap, suggesting that once overlap is scarce, support and source genre dominate. Valerius draws on well-attested poetic epic, whereas Jerome's sparse allusions to prose sources such as Cicero are effectively missed. The few near-perfect verbatim scores (F1 = 1.00 for Jerome–Lucan, Jerome–Ovid, and Lactantius–Horace) rest on minimal support (≤ 5 pairs each) and should be read with caution.

Query	Source	Verbatim (cit.)			Allusion (cf.)		
		#	Ov.	F1	#	Ov.	F1
Jerome	Cicero	91	5.22	0.66	21	3.14	0.04
Jerome	Horace	41	6.80	0.98	–	–	–
Jerome	Lucan	2	5.00	1.00	–	–	–
Jerome	Ovid	2	6.00	1.00	–	–	–
Jerome	Virgil	211	6.95	0.89	38	3.20	0.12
Lactantius	Horace	5	12.40	1.00	–	–	–
Lactantius	Lucretius	24	13.21	0.99	–	–	–
Lactantius	Ovid	23	11.13	0.99	–	–	–
Lactantius	Propertius	3	8.00	1.00	–	–	–
Lactantius	Virgil	78	8.85	0.92	–	–	–
Valerius	Lucan	65	1.91	0.52	85	1.33	0.32
Valerius	Ovid	55	1.86	0.52	94	1.16	0.29
Valerius	Statius	42	2.26	0.49	78	1.70	0.23
Valerius	Virgil	207	2.13	0.53	305	1.57	0.14
Total		849	5.11	0.76	621	1.65	0.11

Table 5: **Per-author-pair performance breakdown for XLM-R Large.** Support (#), mean number of shared lemmas per pair (Ov.), and per-query macro F1 (averaged over 5 folds).

Class	Acc.	Spearman ρ (overlap vs. correct)		
		Shared	Jaccard	Q-cov.
Overall	0.969	−0.09	−0.07	−0.09
No match (gold)	0.969	−0.09	−0.07	−0.08
Verbatim (cit.)	0.839	**0.36**	**0.40**	**0.38**
Allusion (cf.)	0.675	$0.04^{\dagger}$	$-0.01^{\dagger}$	$0.01^{\dagger}$

Table 6: **Correlation between lexical overlap and per-instance correctness**, by gold class, for XLM-R Large on the full evaluation grid. Overlap features are shared tokens, Jaccard, and query coverage (Q-cov.). All values significant at $p < 0.001$ except those marked †.

At the instance level, the effect is strongly class-dependent (Table 6). For verbatim references, lexical overlap predicts correctness strongly (Spearman $\rho = 0.36$–0.40, $p < 0.001$); for paraphrased allusions the correlation collapses to near zero and is not significant; and for pairs labeled *no match* it turns negative ($\rho \approx -0.09$), confirming that high overlap encourages spurious matches, the quantitative counterpart of the *omnia timeo* false positive discussed below. The weakly negative overall correlation is therefore an artifact of class balance: pairs labeled *no match* dominate the pair grid, not evidence that overlap is harmful in general.

6.2 Qualitative Analysis

Inspecting individual failures highlights two recurring patterns. In a representative false positive, Jerome's generic *omnia timeo* ("I fear everything") overlaps with a Ciceronian passage, but his cue *ut ait gentilis poeta* ("as the pagan poet says") in fact points to Virgil's *omnia tuta timens*, whose

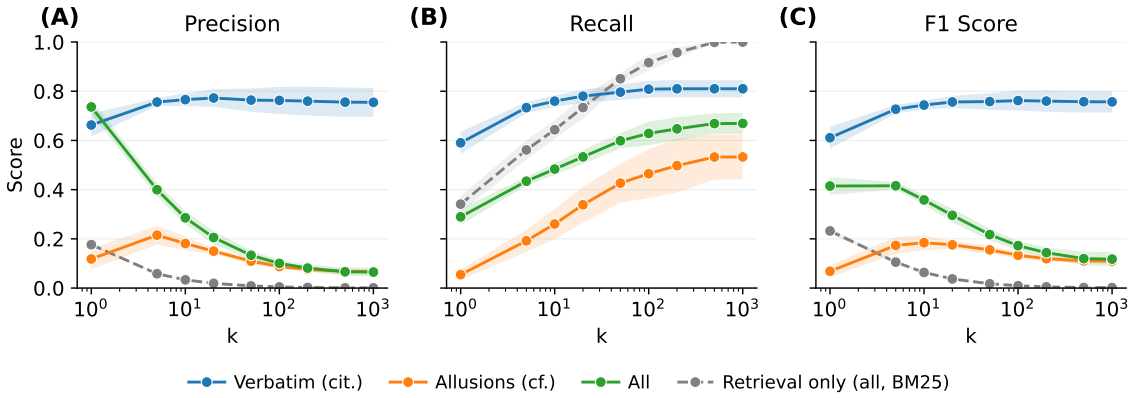


Figure 4: **Performance vs. efficiency trade-off in the retrieve-and-rerank pipeline.** The pipeline first retrieves the top-k candidates with BM25 over lemma unigrams, then applies a cross-encoder classifier to assign the labels *no match*, *verbatim reference*, or *allusion*. We compare this against a retrieval-only baseline where the top-k candidates are treated as positive predictions. Results are averaged across 5 folds.

distinctive word *tuta* is absent from the matched candidate: the model fires on a high-overlap distractor rather than the correct, lower-overlap target. In a representative false negative, the brief Socratic allusion *nosce te* ("know yourself") opens a Jerome passage that then shifts to an unrelated discussion of stylistic simplicity, and the pooled sentence embedding loses the local cue. The two failure modes jointly motivate token- or span-level matching for short, low-overlap references. Full text of both examples, source pairs, and philological discussion are in Appendix J.1.

7 Discussion & Conclusion

In this work, we introduced *Loci Similes*, a benchmark dataset for Latin intertextuality detection designed to evaluate the capacity of language models to capture semantic similarity beyond exact lexical matching. Our baseline experiments demonstrate that while dense retrieval effectively identifies long verbatim references and thematic allusions, detecting subtle short parallels, i.e., intertextual links that share only two or three words, remains challenging.

Among the evaluated models, BM25 with lemmatized unigrams achieved the best retrieval performance (Recall@100 of 0.78 for verbatim references) and XLM-R Large the best classification results (verbatim F1 of 0.76). When retrieving the top 100 candidates per query and reranking with the classifier, the retrieve-and-rerank pipeline recovers 63% of true intertextual references while reducing the candidate set by over 96%. Overall, our findings suggest that language models offer a promising avenue for this task, but the primary challenge lies in distinguishing meaningful reuse from coincidental lexical overlap, which will require further advances and the development of dedicated architectures for the detection of intertextualities.

Takeaways. Two findings stand out. First, a lemma-level BM25 baseline outperforms every dense retriever we evaluate: Latin's rich morphology fragments under subword tokenizers, and CLTK lemmatization gives sparse methods an edge, motivating hybrid retrieval. Second, neural cross-encoders show a clear advantage over lexical classifiers on allusions, achieving noticeably higher precision on the subtler, low-overlap cases where lexical features collapse, while remaining competitive on verbatim references. Consistent with this, per-instance correctness correlates strongly with token overlap for verbatim references (Spearman $\rho \approx 0.4$) but the correlation vanishes for allusions, indicating that progress on allusions cannot come from lexical cues alone.

Future Work. Our error analysis suggests two directions for the hardest cases. First, since pooled bi-encoder embeddings lose the faint signal of short, low-overlap allusions, token-level matching (e.g., late-interaction retrievers) and objectives rewarding semantic and thematic correspondence over lexical overlap may help. Second, because signals are often locally present yet embedded in divergent context, matching at the span or clause level may recover allusions that current methods discard. A complementary direction is interpretability: probing what distinguishes verbatim reuse from allusion, which lexical, syntactic, or thematic features the models rely on, and whether the citing author's intention behind a reference can be characterized.

Limitations

Data Coverage. Although our dataset comprises expert-verified positive pairs, it is not (and cannot be) exhaustive. The labeled dataset likely omits some valid references between the selected works. Consequently, it is possible that a few instances classified as "false positives" may represent genuine but undocumented intertextual links, potentially skewing the reported precision.

Labeling Ambiguity. Defining what constitutes a "reference" in Classical Philology remains a substantial methodological challenge. The boundaries between literal citation, subtle allusion, and general thematic resonance are fluid and subject to differing scholarly definitions. This inherent ambiguity affects the consistency of manual labeling, as annotators may prioritize differing criteria for intertextuality. While the taxonomy in Figure 3 provides a useful analytical framework, it does not apply uniformly across all instances of intertextuality. Many cases exhibit characteristics of multiple categories, and only a minority can be unambiguously assigned to a single class.

AI Usage Statement

Language model-based AI tools (Codex and GitHub Copilot) were used as coding assistants during implementation. Additionally, AI-based tools were used for grammar and spell checking. No AI-generated text appears in the final manuscript.

References

Parul Awasthy, Aashka Trivedi, Yulong Li, Mihaela A. Bornea, David Cox, Abraham Daniels, Martin Franz, Gabe Goodhart, Bhavani Iyer, Vishwajeet Kumar, Luis A. Lastras, J. Scott McCarley, Rudra Murthy, Vignesh P, Sara Rosenthal, Salim Roukos, Jaydeep Sen, Sukriti Sharma, Avirup Sil, Kate Soule, Md. Arafat Sultan, and Radu Florian. 2025. Granite embedding models. *CoRR*, abs/2502.20204.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. *CoRR*, abs/2009.10053.

Arianna Di Bernardo, Simone Poetto, Pietro Sillano, Beatrice Villata, Weronika Sójka, Zofia Pietka-Danilewicz, and Piotr Pranke. 2021. Latin writing styles analysis with machine learning: New approach to old questions. *CoRR*, abs/2109.00601.

Patrick J. Burns, James Brofos, Kyle Li, Pramit Chaudhuri, and Joseph P. Dexter. 2021. Profiling of intertextuality in latin literature using word embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4900–4907. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Joseph P. Dexter, Pramit Chaudhuri, Patrick J. Burns, Elizabeth D. Adams, Thomas J. Bolt, Adriana Cásarez, Jeffrey H. Flynt, Kyle Li, James F. Patterson, Ariane Schwartz, and Scott Shumway. 2024. A Database of Intertexts in Valerius Flaccus; Argonautica 1: A Benchmarking Resource for the Evaluation of Computational Intertextual Search of Latin Corpora;. *Journal of Open Humanities Data*, 10(1).

Stefan Daniel Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4324–4328. Association for Computational Linguistics.

Ashley Gong, Katy Gero, and Mark Schiefsky. 2025. Augmented close reading for classical Latin using BERT for intertextual exploration. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 403–417, Albuquerque, USA. Association for Computational Linguistics.

Vanessa B. Gorman and Robert J. Gorman. 2016. Approaching questions of text reuse in ancient greek using computational syntactic stylometry. *Open Linguistics*, 2(1).

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742.

Maciej Janicki, Antti Kanner, and Eetu Mäkelä. 2023. Detection and attribution of quotes in finnish news media: BERT vs. rule-based approach. In *Proceedings of the 24th Nordic Conference on Computational*

Linguistics, NoDaLiDa 2023, Tórshavn, Faroe Islands, May 22-24, 2023, pages 52–59. University of Tartu Library.

A. J. Kleywegt. 2005. *Valerius Flaccus, Argonautica, Book I.* Mnemosyne, Bibliotheca Classica Batava. Supplementum. Brill, Leiden.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Comput. Linguistics*, 48(4):949–986.

Pak Kin Lau and Stuart Michael McManus. 2024. Mining asymmetric intertextuality. *CoRR*, abs/2410.15145.

Ansel MacLaughlin, Shaobin Xu, and David A. Smith. 2021. Recovering lexically and semantically reused texts. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, *SEM 2021, Online, August 5-6, 2021*, pages 52–66. Association for Computational Linguistics.

Ananth Mahadevan, Michael Mathioudakis, Eetu Mäkelä, and Mikko Tolonen. 2025. Text reuse in large historical corpora: insights from the optimization of a data science system. *Int. J. Data Sci. Anal.*, 20(5):4631–4643.

Enrique Manjavacas, Folgert Karsdorp, and Mike Kestemont. 2020. A statistical foray into contextual aspects of intertextuality. In *Proceedings of the Workshop on Computational Humanities Research (CHR 2020), Amsterdam, The Netherlands, November 18-20, 2020*, volume 2723 of *CEUR Workshop Proceedings*, pages 77–96. CEUR-WS.org.

Enrique Manjavacas, Brian Long, and Mike Kestemont. 2019. On the feasibility of automated detection of allusive text reuse. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, LaTeCH@NAACL-HLT 2019, Minneapolis, MN, USA, June 7, 2019*, pages 104–114. Association for Computational Linguistics.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn J. Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *CoRR*, abs/2509.06888.

Hadar Miller, Tsvi Kuflik, and Moshe Lavee. 2025. Text alignment in the service of text reuse detection. *Applied Sciences*, 15(6).

Fynn Petersen-Frey and Chris Biemann. 2024. Fine-grained quotation detection and attribution in german news articles. In *Proceedings of the 20th Conference on Natural Language Processing, KONVENS 2024, Vienna, Austria, September 10-13, 2024*, pages 196–208. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Marie Revellio. 2022. *Zitate der ›Aeneis‹ in den Briefen des Hieronymus: Eine digitale Intertextualitätsanalyse zur Untersuchung kultureller Transformationsprozesse*. De Gruyter, Berlin, Boston.

Frederick Riemenschneider and Anette Frank. 2023a. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15181–15199. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023b. Graecia capta ferum victorem cepit. detecting latin allusions to ancient greek literature. In *Proceedings of the Ancient Language Processing Workshop, ALP@RANLP 2023, Varna, Bulgaria, 8 September, 2023*, pages 30–38. INCOMA Ltd., Shoumen, Bulgaria / ACL.

David Rosson, Eetu Mäkelä, Ville Vaara, Ananth Mahadevan, Yann Ciarán Ryan, and Mikko Tolonen. 2023. Reception reader: Exploring text reuse in early modern british publications. *CoRR*, abs/2302.04084.

Franziska Schropp, Thomas E. Konrad, Marie Revellio, and Barbara Feichtinger. 2024a. Towards a differentiated digital-hermeneutic analysis tool for the detection of short quotations using the example of the church father jerome. *Digit. Humanit. Q.*, 18(3).

Franziska Schropp, Michael Wittweiler, Thomas E. Konrad, Marie Revellio, and Barbara Feichtinger. 2024b. Manuell-hermeneutische forschung als goldstandard: Zur kalibrierung digitaler analysetools in den digital humanities. *Digital Classics Online*, 10.

Muhammad Sharjeel, Iqra Muneer, Sumaira Nosheen, Rao Muhammad Adeel Nawab, and Paul Rayson. 2023. Cross-lingual text reuse detection at document level for english-urdu language pair. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 22(6):173:1–173:22.

François Spaltenstein. 2002. *Commentaire des Argonautica de Valérius Flaccus (livres 1 et 2)*. Éditions Latomus, Bruxelles.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1102–1121. Association for Computational Linguistics.

Timothy R Tangherlini and Ruofei Chen. 2024. Travels with bert: Surfacing the intertextuality in hans christian andersen's travel writing and fairy tales through the network lens of large language model-based topic modeling. *Orbis Litterarum*, 79(6):519–562.

Hawau Olamide Toyin, Federico Iezzi, Elia Scapini, Giulio Federico, and Giovanni Puccetti. 2026. Gretino: A greek and latin dataset to benchmark retrieval systems in classical languages. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 919–928, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Tedo Vrbanec and Ana Mestrovic. 2023. Comparison study of unsupervised paraphrase detection: Deep learning - the key for semantic similarity detection. *Expert Syst. J. Knowl. Eng.*, 40(9).

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 952–963. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 text embeddings: A technical report. *CoRR*, abs/2402.05672.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 2526–2547. Association for Computational Linguistics.

Yuanchi Zhang and Yang Liu. 2022. Directquote: A dataset for direct quotation extraction and attribution in news articles. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 6959–6966. European Language Resources Association.

Andrew Zissos. 2008. *Valerius Flaccus' Argonautica Book I: Edited with Introduction, Translation, and Commentary*. Oxford University Press, Oxford.

A Corpus Sources

The corpus texts were aggregated from three digital repositories: *Corpus Corporum*[4], the *Tesserae Project*[5], and the *OpenGreekandLatin Project*. Table 7 lists the editions used; Figure 13 illustrates representative intertextual links from the ground truth.

Author	Work	Source	Edition
Virgil	Aeneid	CC	Greenough (1900)
Virgil	Georgics	CC	Greenough (1900)
Virgil	Eclogues	CC	Greenough (1900)
Ovid	Amores	CC	Ehwald (1907)
Ovid	Ars Amatoria	CC	Ehwald (1907)
Ovid	Ex Ponto	CC	Wheeler (1939)
Ovid	Fasti	CC	Frazer (1933)
Ovid	Heroides	CC	Ehwald (1907)
Ovid	Ibis	CC	Merkel/Ehwald (1889)
Ovid	Medicamina	CC	Ehwald (1907)
Ovid	Metamorphoses	CC	Magnus (1892)
Ovid	Remedia Amoris	CC	Ehwald (1907)
Ovid	Tristia	CC	Wheeler (1939)
Martial	Epigrammata	CC	Heraeus (1925)
Lucretius	De Rerum Natura	CC	Martin (1934)
Lucan	Pharsalia	CC	Weise (1835)
Horace	Carmen Saeculare	CC	Shorey (1898)
Horace	Carmina	CC	Shorey (1919)
Horace	Ars Poetica	CC	Smart (1836)
Horace	Epistulae	CC	Fairclough (1929)
Horace	Epodes	CC	Vollmer (1912)
Horace	Saturae	CC	Smart (1836)
Catullus	Carmina	Tess	Merrill
Propertius	Elegiae	Tess	Mueller (1898)
Tibullus	Elegiae	Tess	Postgate (1915)
Cicero	*Opera Omnia*	Tess	*Varia*
Jerome	Epistulae	OGL	Hilberg (1910)
Jerome	*Varia*	CC	Patrologia Latina (1845)
Valerius Flaccus	Argonautica	CC	Kramer (1913)
Statius	Thebais	CC	Mozley (1928)

Table 7: **Corpus data sources.** Abbreviations: CC (Corpus Corporum), Tess (Tesserae Project), and OGL (OpenGreekandLatin Project).

B Evaluation Metrics

The classification model and the full retrieve-and-rerank pipeline are evaluated on the same directional comparison task between *query* and *source* documents, using 5-fold cross-validation over the 1,490 verified positive pairs.

Unlike standard information retrieval tasks that focus on ranking top-k candidates, our objective is to classify the full set of possible links between the query and source documents. Because many query segments have no true positive match in the source document, evaluation must reward the correct rejection of non-links as well as the recovery of attested references. We therefore define $N = \text{TP} + \text{FP} + \text{FN} + \text{TN}$ as the total number of text-segment pairs and use the following error-based metrics, which better match the practical constraints of philological workflows.

[4] https://mlat.uzh.ch/

[5] https://github.com/tesserae/tesserae

- **Segment-Misclassification Rate (SMR):** defined as the fraction of all query-source pairs that were misclassified. This serves as a global error rate. Values range from 0 (perfect retrieval) to 1 (complete failure).

$$\mathrm{SMR} = \frac{\mathrm{FP} + \mathrm{FN}}{N}$$

- **Global False-Positive Rate (FPR):** defined as the share of the total dataset incorrectly predicted as links. A high FPR indicates a system prone to "over-generating" candidate links.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N}$$

- **Global False-Negative Rate (FNR):** defined as the share of the total dataset that contains true links missed by the system. A high FNR indicates that genuine intertextual references remain undiscovered.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{N}$$

Together, these metrics decompose the total error (SMR) into false-positive and false-negative components, making it clear whether a model tends to over-generate candidate links or miss genuine references. We calculate each metric per query segment and report the mean across queries.

C Data Split and Evaluation Setup

All retrieval, classification, and combined-pipeline experiments use the same 5-fold split over the 1,490 verified positive pairs. The split is stratified at the query level: all positive targets of a given query segment are assigned to the same fold, so no query appears in both training and held-out evaluation. Fold sizes therefore vary slightly, because individual queries can cite different numbers of source segments, even though the number of query segments per fold is approximately balanced.

Document-level, fully pairwise evaluation. Each fold simulates the realistic philological setting of comparing two full documents. The *query document* contains held-out query segments plus an equal number of query-side distractors drawn from the same work but without annotated links; the *source document* contains the ground-truth source segments plus an equal number of source-side distractors drawn from the cited works. We then score *every* query segment against *every* source segment in that fold. The evaluation is therefore fully pairwise rather than restricted to gold-aligned pairs, forcing models to distinguish true references from in-domain distractors written by the same authors and in similar registers.

Per-fold sizes. Per-fold sizes (queries, sources, ground-truth pairs, and total pairwise comparisons) are reported in Table 4 in Section 4.4.

The same per-fold query and source documents are reused across retrieval models, classification models, and end-to-end runs, so the reported numbers are directly comparable. On average, each fold requires roughly 1.6×10^5 pairwise scores, of which only $\sim 0.19\%$ are true positives; the global error-rate metrics in Appendix B make this imbalance visible.

D Retrieval Setup

The candidate-retrieval setup covers model families, ablation design, negative sampling, and implementation details. Retrieval results are reported in Appendix G.

D.1 Model Families

We group the retrieval methods by the kind of signal they exploit, from sparse lexical overlap to fine-tuned dense bi-encoders.

Sparse retrievers. We evaluate TF-IDF and BM25 over both surface tokens and CLTK lemmas, using either unigrams alone or unigrams plus bigrams. For each query segment, source segments are ranked by sparse vector similarity (TF-IDF) or BM25 score, without learned parameters. TF-IDF uses a custom analyzer over the CLTK-tokenized input with `sublinear_tf` = True, `max_features` = 50k, `min_df` = 1, and `max_df` = 1.0, fit on the training fold only.

Word-embedding retrievers. We reimplement two prior approaches that operate on word-level rather than sentence-level embeddings. Following Burns et al. (2021), we score segment pairs by the mean cosine similarity of their bigram Word2Vec vectors, testing a Bamman–Burns lemma Word2Vec checkpoint[6] and a LiLa fastText skip-gram checkpoint.[7] Following Gong et al.

[6] `qcl_bamman_lemma_300`
[7] `lila_fasttext_skip_win5_min5`

(2025), we use position-aggregated contextual token embeddings from LatinBERT[8] as the similarity signal.

Dense retrievers. For sentence-level embedding, we evaluate the multilingual E5 family (Small, Base, and Large) (Wang et al., 2024), the Granite embedding models (107m and 278m) (Awasthy et al., 2025), and BGE-M3 (Chen et al., 2024). We also include SPhilBerta (Riemenschneider and Frank, 2023a) as a domain-adapted model pretrained on classical languages. All dense retrievers are used as bi-encoders that embed queries and sources independently and rank candidates by cosine similarity.

D.2 Ablation Design

We evaluate model choice and training regime for the retrieval stage independently before combining the selected components in the end-to-end pipeline.

- **Base-model ablations.** We compare sparse lexical retrievers, word-embedding baselines, and fine-tuned bi-encoders to isolate the effect of lexical matching, word-level semantics, and sentence-level dense retrieval.
- **Negative-sampling ablations.** For the learned bi-encoder, we compare multiple strategies for constructing negatives and sweep the full negative-to-positive training ratio grid 1:1, 1:2, . . . , 1:10 (ten ratios) to measure sensitivity to class imbalance.
- **Hyperparameter ablations.** For the fine-tuned bi-encoder, we sweep learning rate $\in \{1\times10^{-5}, 2\times10^{-5}, 3\times10^{-5}\}$ and number of epochs $\in \{1, 2, 4, 6\}$ (12 configurations $\times$ 5 folds $=$ 60 runs) to identify stable operating regions rather than relying on a single default setting.
- **Pipeline-depth ablations.** For the full retrieve-and-rerank system, we vary retrieval depth over 15 values, from $k = 1$ to $k = 1000$ (1, 5, 10, 20, 50, and then 100–1000 in steps of 100), to measure the trade-off between recall, false positives, and the number of candidate pairs inspected or reranked. A retrieval-only baseline assigns $p = 1.0$ to the top-k candidates and $p = 0.0$ to all remaining pairs, using a 0.5 decision threshold; the reranker is applied only to the top-k pool. These results are reported in Appendix I.

D.3 Negative Sampling

The quality of the learned bi-encoder depends on the negative examples used during training. We evaluated four negative sampling strategies:

- **Random pairs ($\langle$random, random$\rangle$):** Pairs are formed by selecting two completely disjoint segments at random from the corpus.
- **Random negatives ($\langle$query, random$\rangle$):** For each positive query, we sample a negative candidate uniformly at random from the remaining corpus.
- **Hard negatives ($\langle$query, similar$\rangle$):** We use a pre-trained embedding model to identify "hard negatives", candidates that are semantically similar to the query but are not true intertextual references.
- **Mixed negatives ($\langle$query, mix$\rangle$):** We combine uniformly sampled negatives with semantically similar hard negatives for the same query, testing whether a mixture of easy and difficult distractors improves the learned boundary.

D.4 Implementation Details

We evaluate the sparse retrievers (TF-IDF, BM25) over both surface tokens and CLTK lemmas, with unigrams alone and unigrams plus bigrams. All vectorizers share the CLTK tokenization and lemmatization pipeline and are fit on the training fold only.

For the dense retriever, we employ sentence-transformer bi-encoders that generate embeddings for query and source segments independently. During inference, the bi-encoder computes cosine similarity between the query embedding and every source embedding in the index, and candidates are ranked by this score. To adapt the retriever to the task, we fine-tune the model using Online Contrastive Loss, which pulls linked pairs together and pushes unrelated pairs apart.

We fine-tuned the bi-encoder for 4 epochs (batch size 32, learning rate 2×10^{-5}, AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, `max_grad_norm` $= 1.0$, weight decay 0.01) using a linear learning-rate scheduler with 10% warmup and, following Su et al. (2023), prepend "Query: " to query segments and

[8] `ashleygong03/bamman-burns-latin-bert`

"Candidate: " to source segments before encoding. Evaluation during training uses batch size 8 and is run every 50 optimizer steps; we use a fixed seed of 42 across folds. The bigram Word2Vec retriever is trained on the Bamman–Burns Latin corpus.

E Classification Setup

The multiclass reranking setup covers model families, ablation design, sampling strategies, and implementation details. The reranker assigns one of three labels, $0 =$ *no match*, $1 =$ *cit.*, or $2 =$ *cf.*, to each (query, candidate) pair. Classification results are reported in Appendix H.

E.1 Model Families

We group the classifiers by the kind of signal they exploit.

Sparse classifiers. As lexical reference points for reranking, we train a logistic regression and a histogram gradient-boosted decision tree (GBDT) on pairwise similarity features computed for each (query, candidate) pair. These models test how far explicit lexical evidence alone can separate *no match*, *verbatim reference*, and *allusion*.

Neural cross-encoders. For reranking, we evaluate multilingual and domain-adapted transformer encoders as multiclass cross-encoders. The multilingual baselines include XLM-R (Base and Large) (Conneau et al., 2020), ModernBERT (Base and Large) (Warner et al., 2025), and mmBERT (Small and Base) (Marone et al., 2025). To measure the impact of domain adaptation, we evaluate PhilBerta (Riemenschneider and Frank, 2023a), LaBerta, and RoBERTa-Latin, which are pre-trained on Latin corpora. We also include BERT-Romanian (Dumitrescu et al., 2020) to test cross-lingual transfer from a related Romance language. Unlike the bi-encoder retrievers, cross-encoders process the query and candidate jointly and predict one of the three relation labels directly.

E.2 Ablation Design

We evaluate model choice and training regime for the reranking stage independently before combining the selected components in the end-to-end pipeline.

- **Base-model ablations.** We compare sparse lexical classifiers against multilingual and domain-adapted transformer cross-encoders.
- **Negative-sampling ablations.** We compare multiple strategies for constructing negatives and sweep the full negative-to-positive training ratio grid $1{:}1, 1{:}2, \ldots, 1{:}10$ (ten ratios) to measure sensitivity to class imbalance. Pair labels are looked up in a precomputed (query, corpus) table; when a pair carries both a *cit.* and a *cf.* annotation, *cit.* takes precedence.
- **Three-way class-sampling ablations.** In a follow-up experiment for the 3-class classifier, we vary the target sampling proportions of the three training labels, $0 =$ *no match*, $1 =$ *cit.*, and $2 =$ *cf.*, using a target-class-probability sampler. The default rerun uses a mild fixed exposure of $\{0{:}0.80, 1{:}0.10, 2{:}0.10\}$; the ablation also tests $\{0.50, 0.25, 0.25\}$, $\{0.60, 0.20, 0.20\}$, $\{0.33, 0.33, 0.33\}$, and a "natural" setting without resampling. This tests how strongly the reranker depends on the relative frequency of negatives, verbatim references, and allusions during training.

E.3 Negative Sampling and Class Sampling

The quality of the multiclass reranker depends on the negative examples and class-mixing strategy used during training. We evaluated four negative sampling strategies:

- **Random pairs ($\langle$random, random$\rangle$):** Pairs are formed by selecting two completely disjoint segments at random from the corpus.
- **Random negatives ($\langle$query, random$\rangle$):** For each positive query, we sample a negative candidate uniformly at random from the remaining corpus.
- **Hard negatives ($\langle$query, similar$\rangle$):** We use a pre-trained embedding model to identify "hard negatives", candidates that are semantically similar to the query but are not true intertextual references.
- **Mixed negatives ($\langle$query, mix$\rangle$):** We combine uniformly sampled negatives with semantically similar hard negatives for the same query, testing whether a mixture of easy and difficult distractors improves the learned boundary.

In addition to choosing how negatives are constructed, the multiclass reranker requires a decision about how often each of the three labels is

shown during training. After remapping the labels to $0 =$ *no match*, $1 =$ *cit.*, and $2 =$ *cf.*, training batches are drawn with a target-class-probability sampler rather than relying only on the raw empirical class frequencies. The default setting uses class probabilities $\{0{:}0.80, 1{:}0.10, 2{:}0.10\}$, and the ablation varies these three-way proportions to measure the effect of emphasizing negatives versus the two positive reference types during training.

E.4 Implementation Details

The sparse classifiers (logistic regression and GBDT) use pairwise features for each (query, candidate) pair: TF-IDF cosine similarities over lemma unigrams, lemma unigrams+bigrams, and character 3–4-grams, plus Jaccard overlap on lemmas, raw token overlap, and length features. All vectorizers share the retrievers' CLTK tokenization and lemmatization pipeline and are fit on the training fold only.

For the cross-encoder, we concatenate query and candidate, separated by the model's special tokens (Figure 11), and truncate each to 50% of the token budget to prevent either from dominating.

<s> *Legimus ...* </s></s> *excudent...* </s>

We fine-tuned the cross-encoder for 4 epochs (per-GPU batch size 32 for base models and 16 for large models, with gradient accumulation chosen to keep an effective batch size of 64 across GPUs, learning rate 2×10^{-5}, AdamW) to predict the multiclass relation label for each query-candidate pair and minimize cross-entropy against the ground-truth labels. RoBERTa-family encoders use an explicit separator fix to insert the correct </s></s> sequence between query and candidate.

In the 3-class classifier follow-up, the cross-encoder uses a target-class-probability sampler during training. This sampler is implemented as a `WeightedRandomSampler` that is re-drawn each epoch, so that the empirical class exposure matches the configured target within each fold. We do not additionally re-weight the cross-entropy loss, so the sampler is the only mechanism used for the sampling-proportion ablation described above.

For reporting, per-class decision thresholds are tuned on the training fold using a *plateau-high* heuristic with tolerance 0.01: starting from the highest threshold, we walk down the precision–recall curve and pick the largest threshold at the top of the F1 plateau, independently for the *cit.* and *cf.* classes. The remaining mass is assigned to *no match*. These tuned thresholds are then frozen and applied to the held-out fold.

F Combined Pipeline Setup

The combined experiment evaluates the full retrieve-and-rerank system on the fold-level evaluation files described in Appendix C. For each held-out fold, the retriever first scores every query segment against the source document and returns the top-k candidates; the reranker then classifies only this candidate pool. All remaining query–source pairs are treated as negative predictions. This preserves the fully pairwise document-level evaluation while measuring how much of the exhaustive comparison grid can be avoided through candidate generation.

The final combined configuration uses BM25 with lemma unigrams (the strongest sparse retriever; see Appendix G) as the candidate generator. The reranking stage uses the fine-tuned XLM-R Large multiclass cross-encoder from the classification experiments. For each candidate pair, the reranker predicts $0 =$ *no match*, $1 =$ *cit.*, or $2 =$ *cf.*; the two positive classes are merged when reporting reference-detection performance, and ref-type-specific metrics are computed by filtering the same cached predictions.

We compare this retrieve-and-rerank system against a retrieval-only baseline at the same retrieval depths. The retrieval-only baseline assigns all top-k candidates to the positive class and all other pairs to *no match*; the combined pipeline instead lets the classifier decide which retrieved candidates remain positive. Retrieval depth is swept over 15 values, from $k = 1$ to $k = 1000$ (1, 5, 10, 20, 50, and then 100–1000 in steps of 100); the reported combined-pipeline results in Appendix I focus on representative depths from this grid.

G Retrieval Results

We evaluate candidate retrieval with standard information-retrieval metrics, focusing on whether the first stage preserves enough true references for downstream reranking. Table 8 and Figure 5 show that the larger multilingual embedding models have the strongest recall. Table 9 reports the lexical and word-embedding retrieval baselines. Figure 6 gives the fold-level behavior of the best-performing model, while Figure 7 and Figure 8 summarize the negative-ratio and hyperparameter ablations.

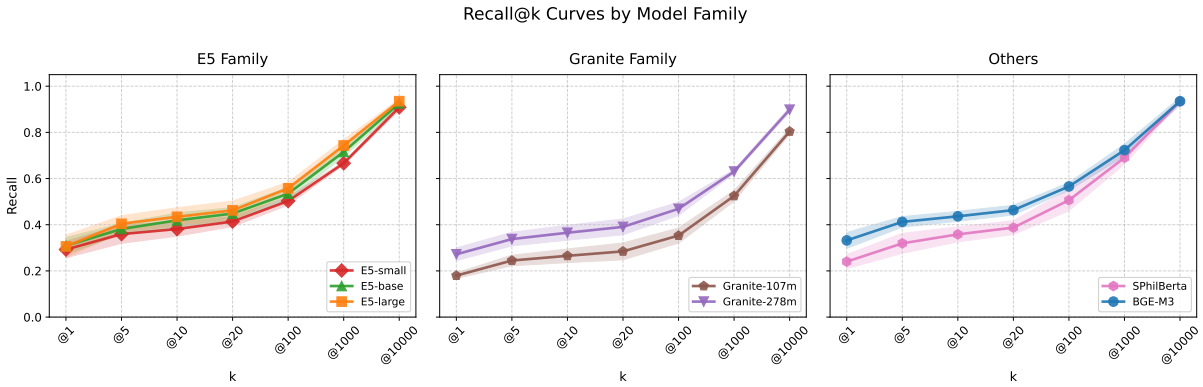


Figure 5: **Recall@k comparison across embedding model families.** Performance of E5, Granite, BGE-M3, and the domain-specific SPhilBerta on the Latin intertextuality retrieval task. We observe that larger models show better performance, with multilingual E5-large achieving the highest scores overall. Error bands show ± 1 std across 5-fold cross-validation.

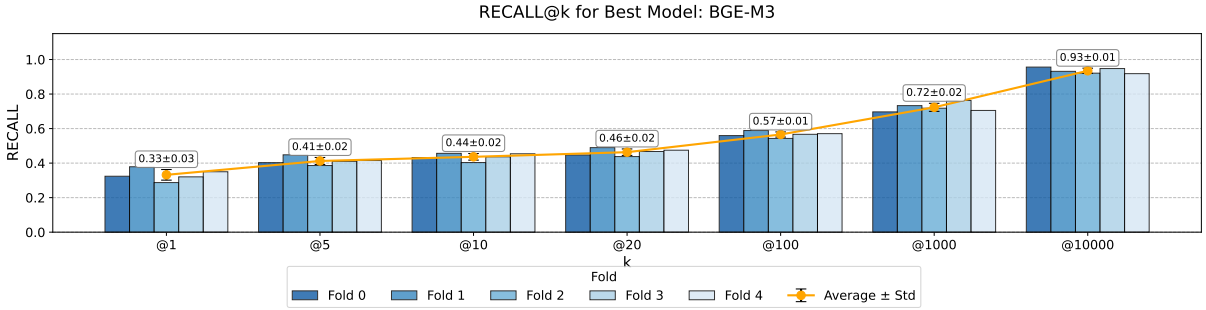


Figure 6: **Recall@k performance of the best embedding model across individual folds.** We show recall at varying cutoff values ($k \in \{1, 5, 10, 20, 100, 1000, 10000\}$) for the top-performing model on the Latin intertextuality retrieval task. Each bar represents a single fold from 5-fold cross-validation, with the orange line indicating mean $\pm$ std. Performance improves substantially with larger k, approaching near-perfect recall at $k = 10000$.

H Classification Results

We evaluate the multiclass cross-encoder with standard classification metrics and the task-specific global error rates defined in Appendix B.

H.1 Sampling and Models

Table 10 shows that query-conditioned random negatives ($\langle$qry, rnd$\rangle$) reduce global error rates more effectively than fully random pairs ($\langle$rnd, rnd$\rangle$). Table 11 reports the 3-class model under different class-sampling proportions. Table 12 compares the base architectures, while Figure 9 summarizes the negative-ratio sweep.

I Combined Pipeline Results

Table 13 compares the retrieval-only baseline with the retrieve-and-rerank pipeline at varying retrieval depths (k), relating performance gains to the number of candidate pairs sent to the classifier. Figure 10 visualizes the same precision–efficiency trade-off.

J Error Analysis

Table 14 reports the full per-author-pair, per-class breakdown for XLM-R Large on the pairwise evaluation grid. Unlike the compact table in the main error-analysis section, this appendix table includes precision, recall, F1, and the locisimiles error-rate metrics (FPR, FNR, SMR) for each author pair and each positive class.

The table should be read as a descriptive diagnostic, not as a pooled confusion matrix over the entire dataset. Each rate metric is computed within each fold using the locisimiles per-query macro definition in a one-vs-rest setup at $p \geq 0.5$, and the reported values are then averaged over folds. Support (#), by contrast, is summed across folds. Rows with very small support can therefore show extreme scores; the goal is to expose variation by author pair and reference type rather than to provide a second aggregate leaderboard.

Model	Recall			MRR		
	@10	@100	@1000	@10	@100	@1000
BGE-M3	**0.436 ± 0.021**	**0.566 ± 0.017**	0.723 ± 0.026	**0.402 ± 0.026**	**0.408 ± 0.025**	**0.408 ± 0.025**
E5-large	0.434 ± 0.039	0.558 ± 0.026	**0.743 ± 0.024**	0.384 ± 0.043	0.391 ± 0.043	0.391 ± 0.043
E5-base	0.419 ± 0.032	0.535 ± 0.028	0.715 ± 0.021	0.377 ± 0.033	0.383 ± 0.032	0.384 ± 0.032
E5-small	0.381 ± 0.031	0.503 ± 0.017	0.666 ± 0.007	0.356 ± 0.029	0.362 ± 0.028	0.362 ± 0.028
Granite-278m	0.365 ± 0.032	0.469 ± 0.029	0.630 ± 0.009	0.327 ± 0.018	0.333 ± 0.017	0.333 ± 0.017
Granite-107m	0.265 ± 0.029	0.353 ± 0.033	0.525 ± 0.023	0.229 ± 0.009	0.233 ± 0.009	0.234 ± 0.009
SPhilBerta	0.358 ± 0.032	0.506 ± 0.046	0.690 ± 0.028	0.303 ± 0.031	0.310 ± 0.030	0.311 ± 0.030

Table 8: **Performance comparison of different base models for the dense retrieval task.** All models were fine-tuned using the same negative sampling strategy. Metrics are averaged across 5 folds.

Model	Recall			MRR		
	@10	@100	@1000	@10	@100	@1000
TF-IDF (surf. 1g)	0.516 ± 0.037	0.652 ± 0.024	0.804 ± 0.019	0.439 ± 0.035	0.446 ± 0.034	0.446 ± 0.034
TF-IDF (lem. 1g)	0.475 ± 0.050	0.623 ± 0.027	0.821 ± 0.014	0.431 ± 0.048	0.438 ± 0.047	0.439 ± 0.046
TF-IDF (lem. 1+2g)	0.494 ± 0.039	0.645 ± 0.040	**0.833 ± 0.021**	0.419 ± 0.037	0.426 ± 0.037	0.426 ± 0.036
BM25 (surf. 1g)	**0.572 ± 0.041**	0.666 ± 0.020	0.799 ± 0.020	**0.556 ± 0.032**	**0.560 ± 0.031**	**0.561 ± 0.031**
BM25 (lem. 1g)	0.545 ± 0.046	**0.680 ± 0.011**	0.824 ± 0.007	0.514 ± 0.056	0.521 ± 0.053	0.522 ± 0.053
Word2vec (WE, 2g)	0.122 ± 0.012	0.264 ± 0.023	0.529 ± 0.033	0.077 ± 0.005	0.083 ± 0.006	0.084 ± 0.006
FastText (WE, LiLa)	0.119 ± 0.019	0.260 ± 0.036	0.491 ± 0.028	0.074 ± 0.011	0.080 ± 0.009	0.081 ± 0.009
LatinBERT (WE)	0.108 ± 0.011	0.147 ± 0.017	0.219 ± 0.024	0.087 ± 0.011	0.089 ± 0.012	0.089 ± 0.012

Table 9: **Performance comparison of baseline retrieval approaches.** Lexical approaches (TF-IDF, BM25) use surface tokens (*surf.*) or lemmas (*lem.*) with unigrams (*1g*) or unigrams+bigrams (*1+2g*); WE denotes approaches based on word embeddings. Metrics are averaged across 5 folds.

Sampling Method	Classification Metrics				Global Error Rates			Confusion Matrix			
	Prec.	Rec.	F1	Acc.	FPR	FNR	SMR	TP	FP	FN	TN
Mixed negatives $\langle \text{qry}, \text{mix} \rangle$	**0.03**	0.61	**0.05**	**0.96**	**0.0443**	0.0007	**0.0450**	179	**7084**	114	**152150**
Random pairs $\langle \text{rnd}, \text{rnd} \rangle$	0.01	**0.67**	0.01	0.71	0.2888	**0.0006**	0.2894	**195**	44578	**99**	114661
Random negatives $\langle \text{qry}, \text{rnd} \rangle$	0.02	0.66	0.04	0.94	0.0626	0.0006	0.0633	193	10154	101	149085
Hard negatives $\langle \text{qry}, \text{sim} \rangle$	0.01	0.61	0.02	0.89	0.1100	0.0007	0.1107	179	17554	114	141682

Table 10: **Performance comparison of the classification model across different negative sampling strategies.** Metrics are computed from the thresholded multiclass evaluation. The 3-class confusion matrix is collapsed over the two positive classes (*cit* and *cf*); wrong positive-type predictions count as class-level errors. FPR, FNR, and SMR denote global error rates normalized by the total number of pairs N. Results are averaged across 5 folds.

Sampling Proportion	Classification Metrics				Global Error Rates			Confusion Matrix			
	Prec.	Rec.	F1	Acc.	FPR	FNR	SMR	TP	FP	FN	TN
1 : 1 **neg. {0: 0.50, 1: 0.29, 2: 0.21}**	0.01	0.67	0.02	0.81	0.1917	0.0006	0.1924	195	30148	98	129099
1 : 2 **neg. {0: 0.67, 1: 0.19, 2: 0.14}**	0.01	0.69	0.02	0.83	0.1704	0.0006	0.1709	202	27353	91	131896
1 : 3 **neg. {0: 0.75, 1: 0.14, 2: 0.11}**	0.03	0.64	0.07	0.96	0.0401	0.0007	0.0407	188	6340	106	152910
1 : 4 **neg. {0: 0.80, 1: 0.12, 2: 0.08}**	0.03	**0.70**	0.05	0.94	0.0640	**0.0006**	0.0646	**205**	10643	**88**	148601
1 : 5 **neg. {0: 0.83, 1: 0.10, 2: 0.07}**	0.02	0.68	0.03	0.91	0.0924	0.0006	0.0930	200	14984	93	144253
1 : 6 **neg. {0: 0.86, 1: 0.08, 2: 0.06}**	0.03	0.65	0.06	0.96	0.0396	0.0006	0.0402	191	6177	102	153063
1 : 7 **neg. {0: 0.88, 1: 0.07, 2: 0.05}**	0.02	0.66	0.04	0.94	0.0590	0.0006	0.0596	194	9520	100	149714
1 : 8 **neg. {0: 0.89, 1: 0.06, 2: 0.05}**	0.04	0.66	0.07	0.96	0.0373	0.0006	0.0380	193	5894	101	153345
1 : 9 **neg. {0: 0.90, 1: 0.06, 2: 0.04}**	0.03	0.63	0.06	0.95	0.0480	0.0007	0.0487	184	7491	109	151743
1 : 10 **neg. {0: 0.91, 1: 0.05, 2: 0.04}**	**0.04**	0.61	**0.08**	**0.97**	**0.0335**	0.0007	**0.0342**	179	**5245**	115	**153993**

Table 11: **Performance comparison of the 3-class classification model across different class-sampling proportions.** Metrics are computed from the thresholded multiclass evaluation. The 3-class confusion matrix is collapsed over the two positive classes (*cit.* and *cf.*); wrong positive-type predictions count as class-level errors. FPR, FNR, and SMR denote global error rates normalized by the total number of pairs N. Results are averaged across folds.

Model	Verbatim (cit.)			Allusion (cf.)			Avg. F1
	Prec.	Rec.	F1	Prec.	Rec.	F1	
mmBERT Base	0.75 ± 0.02	0.75 ± 0.02	0.74 ± 0.02	0.09 ± 0.03	0.45 ± 0.08	0.13 ± 0.04	0.43 ± 0.02
mmBERT Small	0.72 ± 0.07	0.77 ± 0.04	0.72 ± 0.06	$\mathbf{0.09 \pm 0.02}$	0.37 ± 0.05	$\mathbf{0.13 \pm 0.03}$	0.42 ± 0.04
XLM-R Large	$\mathbf{0.75 \pm 0.06}$	0.81 ± 0.04	$\mathbf{0.76 \pm 0.04}$	0.07 ± 0.01	0.53 ± 0.09	0.11 ± 0.01	$\mathbf{0.43 \pm 0.02}$
XLM-R Base	0.69 ± 0.10	0.75 ± 0.04	0.69 ± 0.08	0.07 ± 0.02	$\mathbf{0.56 \pm 0.08}$	0.11 ± 0.03	0.40 ± 0.05
mBERT Large	0.71 ± 0.03	0.78 ± 0.03	0.72 ± 0.02	0.08 ± 0.01	0.36 ± 0.08	0.12 ± 0.02	0.42 ± 0.01
mBERT Base	0.69 ± 0.07	0.75 ± 0.03	0.69 ± 0.05	0.08 ± 0.02	0.42 ± 0.02	0.12 ± 0.02	0.41 ± 0.04
BERT-Romanian	0.62 ± 0.03	0.71 ± 0.03	0.63 ± 0.02	0.07 ± 0.03	0.42 ± 0.06	0.11 ± 0.04	0.37 ± 0.02
PhilBerta	0.62 ± 0.09	0.73 ± 0.04	0.64 ± 0.07	0.03 ± 0.01	0.45 ± 0.10	0.05 ± 0.02	0.34 ± 0.04
LaBerta	0.25 ± 0.17	0.66 ± 0.05	0.32 ± 0.15	0.02 ± 0.01	0.24 ± 0.06	0.03 ± 0.01	0.17 ± 0.08
LatinBERT	0.06 ± 0.01	0.57 ± 0.06	0.10 ± 0.02	0.02 ± 0.01	0.17 ± 0.05	0.03 ± 0.01	0.06 ± 0.01
RoBERTa-Latin	0.01 ± 0.00	0.26 ± 0.03	0.02 ± 0.00	0.01 ± 0.00	0.12 ± 0.04	0.02 ± 0.00	0.02 ± 0.00
LogReg (lexical)	0.37 ± 0.06	0.83 ± 0.03	0.46 ± 0.06	0.02 ± 0.01	0.37 ± 0.07	0.04 ± 0.01	0.25 ± 0.03
GBDT (lexical)	0.36 ± 0.04	$\mathbf{0.84 \pm 0.02}$	0.45 ± 0.04	0.02 ± 0.01	0.22 ± 0.07	0.04 ± 0.01	0.24 ± 0.02

Table 12: **Performance comparison of different pre-trained base models used in the classification stage.** Scores are given for *Verbatim* (*cit.*) and *Allusion* (*cf.*); the last column is the macro-averaged F1. Per-class decision thresholds are tuned on the training split of each fold. Values are mean $\pm$ std over 5 cross-validation folds.

	# Predictions		Retrieval Only				Retrieve+Rerank				FPR $\times 10^4$		SMR $\times 10^4$	
k	Ret.	Rer.	TP	FP	FN	F1	TP	FP	FN	F1	Ret.	Rer.	Ret.	Rer.
5	2,824	321	164	2,660	130	0.11	127	193	167	0.42	84.05	6.07	88.01	11.28
10	5,648	503	189	5,459	105	0.06	142	361	152	0.36	172.52	11.30	175.65	16.05
20	11,296	777	215	11,081	79	0.04	157	620	137	0.30	350.14	19.37	352.27	23.66
50	28,240	1,355	250	27,990	44	0.02	176	1,179	118	0.22	884.42	36.76	884.98	40.43
100	56,480	1,901	269	56,211	25	0.01	185	1,715	109	0.17	1776.09	53.41	1775.23	56.78
500	282,400	3,253	294	282,106	0	0.00	197	3,056	97	0.12	8913.52	94.03	8905.31	97.00
1000	318,978	3,347	294	318,684	0	0.00	197	3,150	97	0.12	10000.00	96.80	9990.78	99.76

Table 13: **Comparison between retrieval-only and retrieve-and-rerank (best combined pipeline: BM25 + XLM-R Large).** k denotes the number of most similar candidates retrieved by BM25 that are re-ranked using the classification model. **Ret.** = total top-k candidates retrieved (= $k \times n_{\text{queries}}$); **Rer.** = positive predictions output by the reranker. Results are pooled across citation and allusion reference types and averaged across 5 folds. The re-ranking stage substantially reduces error rates (FPR, SMR), scaled by $\times 10^4$ for readability.

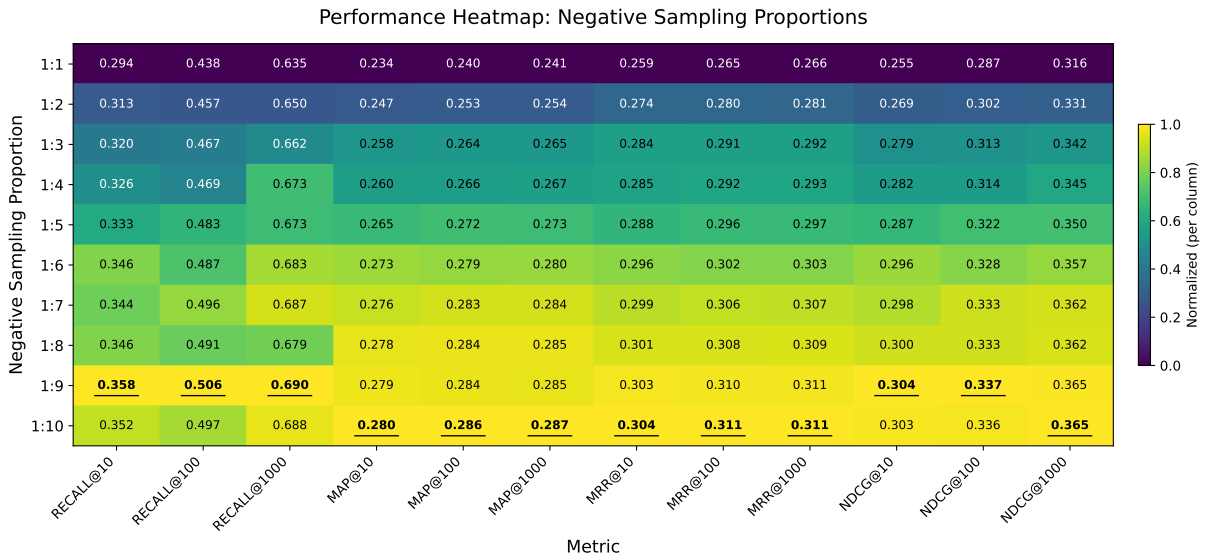


Figure 7: **Impact of training data imbalance on retrieval.** Heatmap showing recall, MAP, MRR, and NDCG scores at different cutoff values (@10, @100, @1000) across negative sampling proportions (1:1 to 1:10). Values are averaged over 5 cross-validation folds. Colors are normalized per column to highlight relative performance differences. Best values per metric are shown in bold with an orange underline. Higher ratios of negative samples generally improve retrieval performance, with optimal results often achieved at 1:5 or higher proportions.

The author-pair breakdown sharpens the pattern from the main text: verbatim performance tracks lexical overlap closely, while allusion performance is weaker and varies more by author pair. Jerome's prose allusions remain especially difficult. The Valerius allusion rows, despite much lower overlap, benefit from larger support and a more homogeneous poetic source domain.

J.1 Qualitative Error Examples

To complement the quantitative breakdown, we inspect two representative cases in detail: a false positive labeled *cit.* and a false negative missed entirely by the model.

False positive. Consider the false positive in Figure 12 (left). The lexical overlap is *omnia timeo* ("I fear everything"), and the verb *timere* ("to fear") recurs in the source as *timenda sint* ("are to be feared"), strengthening the apparent signal. Yet *omnia timeo* is a very general formulation, carrying little distinctive content on its own. The failure is subtler: Jerome writes *ut ait gentilis poeta* ("as the pagan poet says"), signaling a genuine citation, yet it points to Virgil's more specific *omnia tuta timens* ("fearing all things, even safe ones"). The crucial word *tuta* ("safe things"), absent from the matched source, is exactly what makes the Virgilian phrase distinctive. The model thus identifies lexically significant overlap effectively, but it illustrates an edge case requiring expert analysis: the overlap points not to the proposed source but to a different one.

False negative. Inspecting a case missed entirely by the model (Figure 12, right) shows that divergent sentence context can be a problem. The source sentence is wholly concerned with the Socratic saying *nosce te* ("know yourself"). In the query, this reference appears only in the opening imperative *te ipsum intellege!* ("understand yourself!"); the larger part of the query then shifts to a discussion of stylistic simplicity, contrasting *verbosa rusticitas* ("wordy rusticity") with *sancta simplicitas* ("holy simplicity"). The intertextual signal is therefore locally present but embedded in a broader context that points in a different thematic direction, which may explain why the model fails to retrieve the relevant source.

K Python Package

We release the framework described in this paper as an open-source Python package. The `locisimiles`[9] library implements the retrieve-and-rerank pipeline shown in Figure 11 and computes the task-specific error metrics (SMR, FPR, FNR) defined in Appendix B.

Because the package compares all segments of

[9] https://anonymous.4open.science/r/locisimiles-2338/

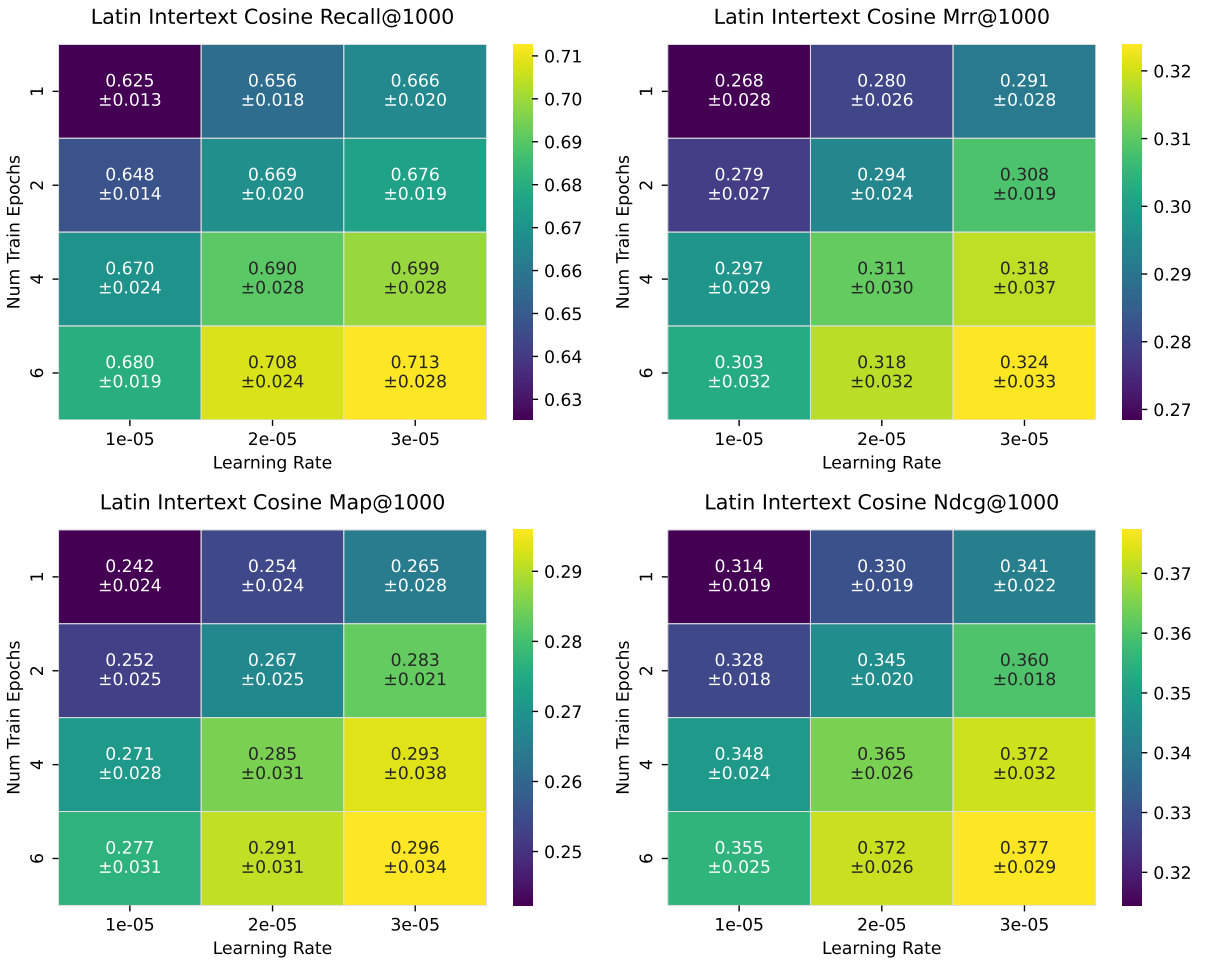


Figure 8: **Hyperparameter sensitivity for the dense retriever.** We visualize retrieval metrics (recall, MRR, MAP, and NDCG at $k = 1000$) across varying learning rates and training epochs. Results are averaged across 5 folds, where annotated values represent the mean score $\pm$ standard deviation.

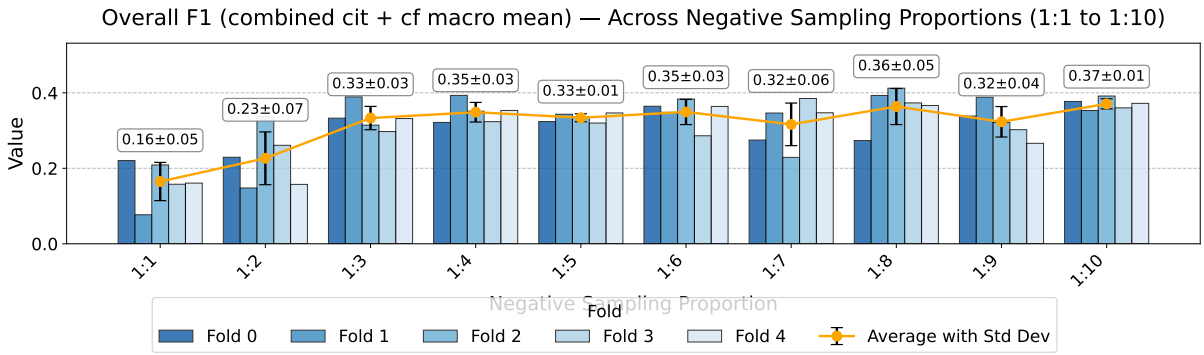


Figure 9: **Impact of training data imbalance on classification performance.** The plot illustrates the recall metric on the evaluation set as the ratio of negative-to-positive training samples is increased from 1:1 to 1:10. Shaded areas represent the standard deviation across the 5 cross-validation folds.

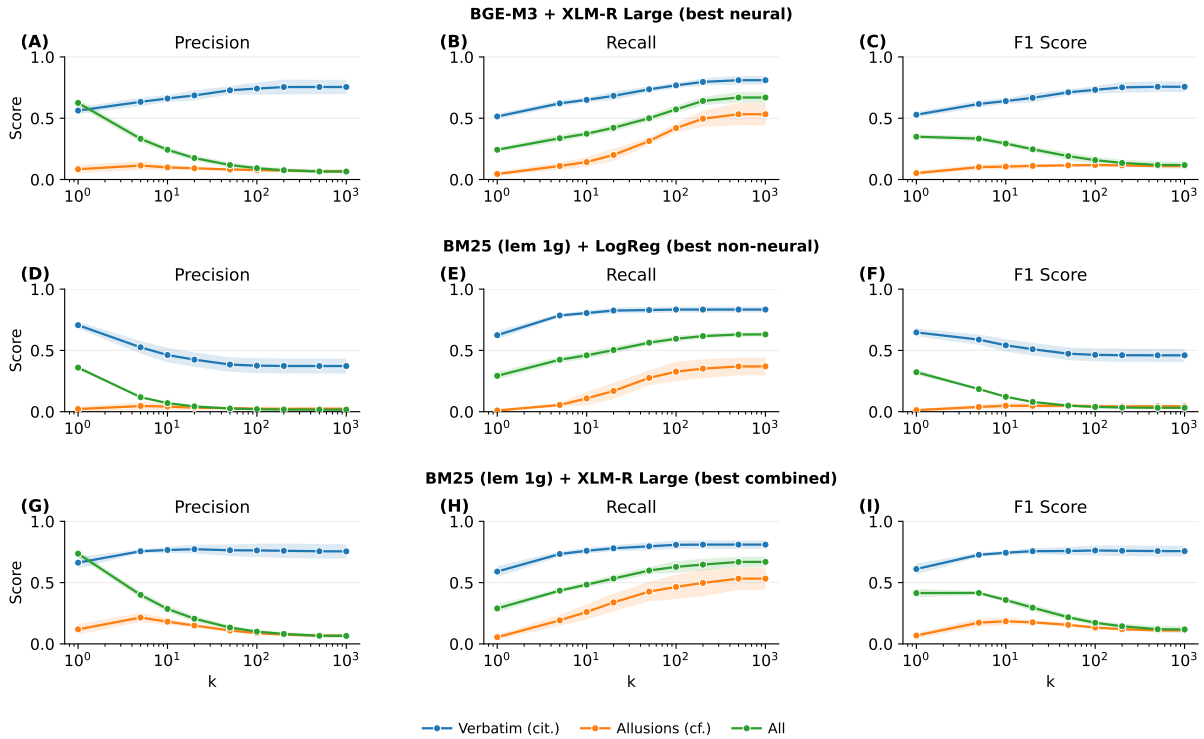


Figure 10: **Precision, Recall, and F1 Score of three retrieve-and-rerank pipeline configurations** as a function of the number of retrieved candidates k. Each row corresponds to one configuration: (*top*) BGE-M3 retrieval + XLM-R Large reranker, (*middle*) BM25 retrieval + Logistic Regression reranker, (*bottom*) BM25 retrieval + XLM-R Large reranker. Within each panel, results are reported separately for verbatim references (*cit.*), allusions (*cf.*), and all references pooled (*all*). The dashed grey curve shows the corresponding *Retrieval only* baseline, in which the top-k retrieved candidates are directly treated as positive predictions without a reranking step. Shaded bands indicate ± 1 standard deviation across 5 cross-validation folds.

Query	Source	Verbatim (cit.)								Allusion (cf.)							
		#	Ov.	P	R	F1	FPR	FNR	SMR	#	Ov.	P	R	F1	FPR	FNR	SMR
Jerome	Cicero	91	5.22	0.651	0.673	0.656	0.003	0.000	0.004	21	3.14	0.050	0.033	0.040	0.001	0.000	0.001
Jerome	Horace	41	6.80	0.980	0.980	0.980	0.003	0.000	0.003	0	–	–	–	–	–	–	–
Jerome	Lucan	2	5.00	1.000	1.000	1.000	0.003	0.000	0.003	0	–	–	–	–	–	–	–
Jerome	Ovid	2	6.00	1.000	1.000	1.000	0.003	0.000	0.003	0	–	–	–	–	–	–	–
Jerome	Virgil	211	6.95	0.878	0.944	0.894	0.004	0.000	0.004	38	3.20	0.077	0.474	0.117	0.017	0.000	0.018
Lactantius	Horace	5	12.40	1.000	1.000	1.000	0.000	0.000	0.000	0	–	–	–	–	–	–	–
Lactantius	Lucretius	24	13.21	0.988	1.000	0.992	0.001	0.000	0.001	0	–	–	–	–	–	–	–
Lactantius	Ovid	23	11.13	0.980	1.000	0.987	0.001	0.000	0.001	0	–	–	–	–	–	–	–
Lactantius	Propertius	3	8.00	1.000	1.000	1.000	0.000	0.000	0.000	0	–	–	–	–	–	–	–
Lactantius	Virgil	78	8.85	0.894	0.983	0.917	0.001	0.000	0.001	0	–	–	–	–	–	–	–
Valerius	Lucan	65	1.91	0.529	0.522	0.518	0.002	0.003	0.005	85	1.33	0.236	0.641	0.324	0.078	0.003	0.081
Valerius	Ovid	55	1.86	0.540	0.511	0.521	0.001	0.001	0.002	94	1.16	0.214	0.609	0.294	0.047	0.002	0.049
Valerius	Statius	42	2.26	0.490	0.500	0.492	0.004	0.003	0.007	78	1.70	0.163	0.570	0.234	0.112	0.004	0.116
Valerius	Virgil	207	2.13	0.552	0.545	0.527	0.002	0.002	0.004	305	1.57	0.089	0.602	0.144	0.086	0.003	0.089
Total		849	5.11	0.755	0.810	0.757	0.003	0.000	0.003	621	1.65	0.067	0.533	0.110	0.017	0.000	0.017

Table 14: **Per-pair, per-class breakdown for XLM-R Large on the full pairwise evaluation set (mean across 5 folds).** Every rate metric (P, R, F1, FPR, FNR, SMR) is computed per fold using the locisimiles per-query macro definition (one-vs-rest at $p \geq 0.5$); support (#) is the sum across folds. Performance metrics (P, R, F1) are shaded teal (darker = higher), error rates (FPR, FNR, SMR) and overlap are shaded so that darker indicates larger values.

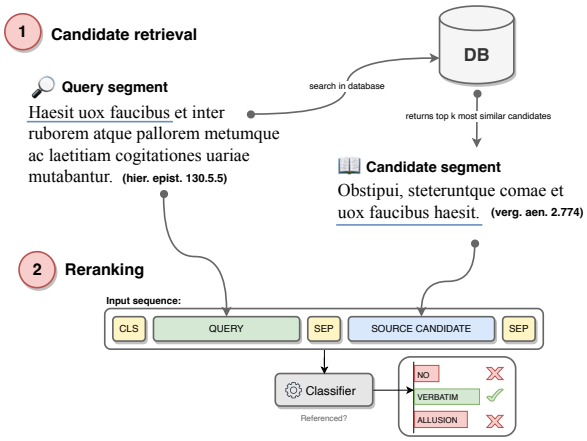


Figure 11: **Retrieve-and-rerank pipeline.** *Stage 1:* The input text segment acts as a query to retrieve potential candidates from the database. *Stage 2:* To verify the reference, the query and source candidate are concatenated into a single input sequence and passed to a three-class cross-encoder (*no match*, *cit.*, *cf.*).

False Positive: Generic Overlap

🔍 **Query** (*Jerome, Epist. 7.4.1*)
LA: *huic ego, ut ait gentilis poeta,* ***omnia*** *etiam tuta* ***timeo****.*
EN: "For this reason, as the pagan poet says, **I fear everything**, even what is safe."

✗ **Predicted source** (*Cicero, Att. 3.8.2*)
LA: *reliqua quam mihi timenda sint video nec quid scribam habeo et* ***omnia timeo****, nec tam miserum est quicquam quod non in nostram fortunam cadere videatur.*
EN: "I see how much I should fear the remaining things, and I do not know what to write; **I fear everything**, and there is nothing so wretched that it does not seem capable of befalling our lot."

✓ **Actual source** (*Virgil, Aen. 4.298*)
LA: *praesensit, motusque excepit prima futuros,* ***omnia tuta timens****.*
EN: "She foresaw it, and was the first to detect the coming disturbances, **fearing everything, even what was safe**."

False Negative: Local Allusion in Broader Context

🔍 **Query** (*Jerome, Epist. 57.12.4*)
LA: ***te*** *ipsum* ***intellege****! uenerationi mihi semper fuit non uerbosa rusticitas, sed sancta simplicitas: qui in sermone imitari se dicit apostolos, prius imitetur in uita.*
EN: "**Understand yourself**! What has always been worthy of reverence to me is not wordy rusticity, but holy simplicity: whoever claims to imitate the apostles in speech should first imitate them in life."

✓ **Source** (*Cicero, Tusc. 1.52*)
LA: *cum igitur* ***nosce te*** *dicit, hoc dicit:* ***nosce animum tuum****.*
EN: "When, therefore, he says '**know yourself**,' he means this: **know your own soul**."

Figure 12: **Qualitative error examples.** Left: false positive labeled *cit.*, where generic *omnia timeo* overlap masks the actual Virgilian source. Right: false negative, where the brief local Socratic allusion *nosce te* is obscured by divergent sentence context.

Examples of Intertextual References

Paraphrase (Minor)

SOURCE
Cicero, Catil. 1.1

"Quo usque tandem **abutere**, Catilina, **patientia nostra**?"
(How long, Catiline, will you **abuse our patience**?)

REUSE
Jerome, Epist. 98.22.4

"... et **patientia nostra** quasi quodam temeritatis fomite **abutentes** ..."
(... and **abusing our patience** like some kindling of rashness ...)

Comment: *Jerome integrates Cicero's famous invective syntactically by adapting the verb form (abutere → abutentes).*

Paraphrase (Major)

SOURCE
Virgil, Georg. 4.82

"... **ingentes animos angusto in pectore** versant ..."
(... they wield **mighty souls in a tiny breast** ...)

REUSE
Jerome, Epist. 107.13.4

"... et **in paruis corpusculis ingentes animos** intueri!"
(... and to see **mighty souls in small bodies**!)

Comment: *Jerome retains the semantic core but rephrases angusto in pectore to in parvis corpusculis.*

Allusion

SOURCE
Cicero, Orat. 33.11

"... sed **nihil difficile amanti** puto."
(... but I think **nothing is difficult for a lover**.)

REUSE
Jerome, Epist. 22.40.1

"**Nihil amantibus durum** est, nullus difficilis cupienti labor."
(**Nothing is hard for lovers**, no labor difficult for the desirous.)

Comment: *Jerome evokes the motif using synonymous but distinct vocabulary (difficile amanti vs. amantibus durum).*

Figure 13: **Example references.** Three instances of text reuse by Jerome included in the ground truth dataset.

the query and source documents, evaluation must account for the correct rejection of non-links as well as the recovery of positive references. In addition to per-class precision (P), recall (R), and F1, `locisimiles` therefore reports three error rates normalized by the total number of segment pairs (N): SMR as a global error rate, FPR for spurious matches, and FNR for missed references. Standard accuracy is omitted because the extreme class imbalance makes it uninformative: a trivial all-negative predictor already achieves near-perfect scores. Formal definitions are given in Appendix B.

K.1 Python API

The core API lets researchers load custom query and source documents in CSV format and run the detection pipeline with pre-trained models from the Hugging Face Hub.

```
# 1. Load query and source documents
query_doc = Document("query.csv")
source_doc = Document("source.csv")

# 2. Initialize the pipeline
pipeline = ClassificationPipeline(
    classification_name="...",
)

# 3. Run the pipeline
results = pipeline.run(
    query=query_doc,
    source=source_doc,
)

# 4. Display results
pretty_print(results)
```

K.2 Graphical User Interface

To lower the barrier to entry, the package includes an optional Gradio-based GUI. It can be installed via the optional dependency group (`pip install . [gui]`) and launched from the command line with `locisimiles-gui`.

Workflow. The application workflow is organized into three sequential stages, as illustrated in Figure 14:

1. **Data Upload:** Users ingest custom query and source documents via CSV files.

2. **Configuration:** The pipeline is customized by selecting pre-trained models and tuning retrieval parameters (e.g., retrieval depth k and classification confidence thresholds).

3. **Result Exploration:** The interactive dashboard presents query segments alongside retrieved source candidates, displaying cosine similarity and classification probability scores, with functionality to export confirmed matches.

The three panels show the upload view (Figure 14A), configuration view (Figure 14B), and result-exploration view (Figure 14C).

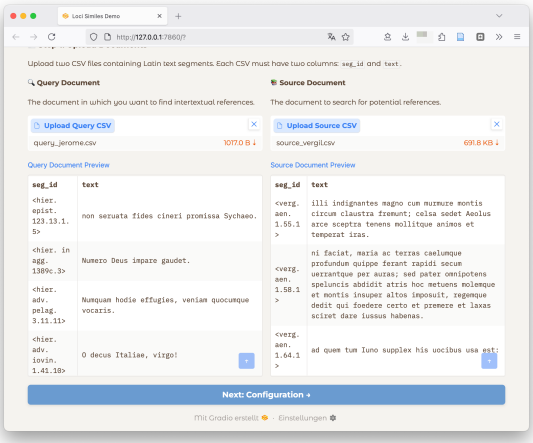
L Scalability and Future Directions

Scalability of the pipeline. The full corpus considered here contains roughly 7.7×10^9 possible query–source pairs (85k query segments against 91k source segments). The retrieval stage reduces this space by orders of magnitude before the cross-encoder is applied, so end-to-end cost is dominated by (i) the dense index built once per source corpus and (ii) cross-encoder forward passes on the top-k candidates per query. At larger corpus scales, neither component remains practical without dedicated indexing: dense retrieval must move from exact to approximate nearest-neighbour search, and the cross-encoder budget must be controlled by reducing k or by precomputing partial token representations. Mahadevan et al. (2025) discuss this billion-scale regime for historical text-reuse detection; their data-structure choices are the closest point of comparison for scaling *Loci Similes* beyond the present setup.

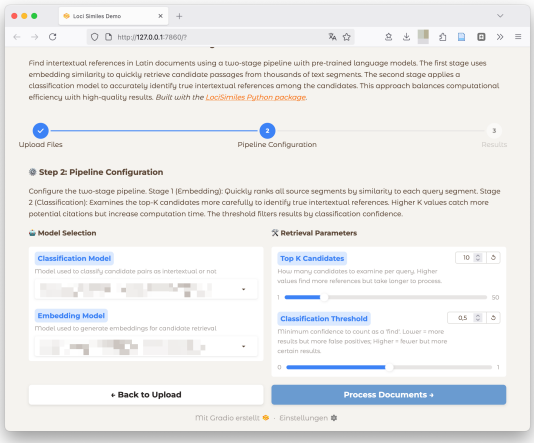
Completeness of the ground truth. As noted in the Limitations of the main body, the expert-verified ground truth is not exhaustive. Some candidate pairs counted here as false positives may correspond to genuine but undocumented intertextual links, so precision figures on the curated subset should be read as lower bounds rather than as estimates over the complete reference graph.

Directions opened by the benchmark. Beyond the modeling directions named in Section 7, the benchmark supports two further lines of work. First, generative explanation: rather than emitting only a score or class label, a model could produce a natural-language rationale for the proposed intertextual link, evaluated against expert annotations for faithfulness and against the qualitative cases in Section 6 for difficulty. Second, indexing-aware retrieval: late-interaction retrievers and the data-structure approaches surveyed by Mahadevan et al. (2025) can be evaluated under the same staged protocol used here, so retrieval quality and retrieval cost can be reported jointly.

(A) Data Upload



(B) Configuration



(C) Interactive Result Exploration

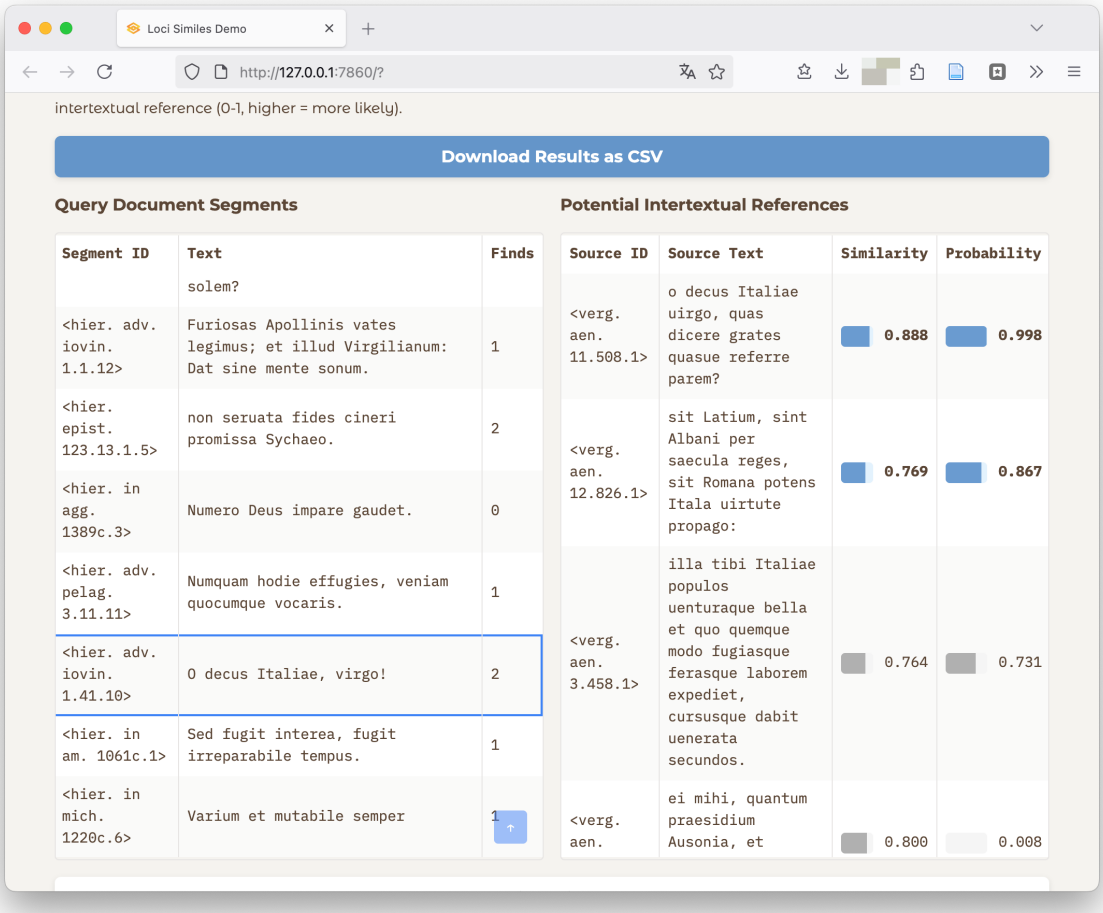


Figure 14: **Graphical user interface workflow.** *(A) Data Upload:* Users ingest query and source documents via CSV files. *(B) Configuration:* The pipeline is customized by selecting pre-trained models and tuning the retrieval depth (top-k) and classification confidence threshold. *(C) Result Exploration:* The dashboard presents query segments alongside retrieved source candidates, displaying both cosine similarity and classification probability scores, with options to export matches as a CSV file.