

MINDMIRROR: DIAGNOSING AND REPAIRING FAILURE MODES IN A HYBRID DENSE-EMBEDDING AND LLM PIPELINE FOR DOCUMENT COMPARISON AND ORIGINALITY ASSESSMENT

Sohail Khan¹, Muhammad Javaid Iqbal^{*1}, Arfan Jaffar², Nouman Aziz³, Saba Ramzan⁴,
Inam Ul Hassan¹

¹Department of Computer Science, The Superior University Lahore, Pakistan

²Faculty of Computer Science, The Superior University Lahore, Pakistan

³Department of Computer Science, University of Gujrat, Pakistan

⁴Department of Computer Science, The University of Lahore, Pakistan

^{*}javaid.ciit@gmail.com, ¹Khanzadasohail067@gmail.com, saba.ramzan@cs.uol.edu.pk,
¹inamulhassan.tech@gmail.com, ²arfan.jaffar@superior.edu.pk, ³noumanaziz1994@gmail.com

DOI: <https://doi.org/10.5281/zenodo.22952209>

Keywords

Cosine similarity, document comparison, information retrieval, large language models, paraphrase detection, plagiarism detection, semantic textual similarity.

Article History

Received: 01 June 2026

Accepted: 23 August 2026

Published: 08 September 2026

Copyright @Author

Corresponding Author: *

Muhammad Javaid Iqbal

Abstract

Combining dense retrieval with a large language model (LLM) makes the creation of such systems straightforward, but it remains difficult to believe that they are accurate. Claims of success on carefully curated benchmarks are reported in the literature, but problems that make-or-break performance in production are not measured. The latter is not the case in this paper, but this research study develops a deployed document-comparison system, MindMirror, consisting of a dense-embedding stage and an LLM stage for qualitative analysis of the comparison results, and systematically identifies and addresses its failure cases in a hand-annotated set of 20 documents over 5 topics, 30 pairs, and a 5-level ordinal similarity scale for diagnosis. Four findings are reported. First, the lexical fallback that is used when the embedding service fails is not only less accurate, but ordinally invalid, showing a minimum adjacent-tier gap of -0.24 points (Spearman 0.842 to 0.955) with the dense score, an issue we attribute to unweighted term frequency, and demonstrate that by suppressing the stop words a sublinear term frequency can recover monotonicity (6.63 points, Spearman 0.842 to 0.955), and by repairing the lexical score using isotonic calibration brings it within 9.44 points ME from the dense score. Interestingly, comparing the two documents directly instead of the two documents under evaluation results in a worse score (-1.63 points), which has direct consequences for small collection retrieval. Second, embedding dimensionality does not correlate to accuracy: in five models on the provider's free tier, the 384-dimension multilingual-light-v3.0 does the best separation of tiers (5.92 points) whereas the 1024-dimension model does the worst (1.40). Third, chunk aggregation is not relevant to single-chunk documents but is crucial for multi-chunk documents, where the widely used max-pooling aggregation

collapses (Spearman 0.548) when the documents share boilerplate. Fourthly, an ensemble of eight LLMs achieves the best single model performance (Spearman 0.9938) with no prior knowledge of the best-performing model, and using the five independent prompts simultaneously reduces latency by 1.37 times (still short of the five-fold optimum provider does not parallelise across a single account). This research study also reports a provider-entitlement mismatch, where only 8 of 18 models advertised were invocable. We believe that reporting a score that resulted from a degraded path, without revealing the degradation, is not a quality trade-off, but a correctness error, and the methodology for creating the diagnostic system used here, the ordinal corpora with adjacent-gap analysis, can be extended to any similarity system that is formed from third-party inference services.

I. INTRODUCTION

Understanding the relationship between two documents lies at the heart of a diverse set of professional applications. An examiner asks a question about the reproduction of a source; an editor asks a question about revisions to reviewers; a lawyer asks which of the ten thousand disclosed files is relevant to a pleading. The interesting cases are those where the wording is different, and the substance is the same, and that's where the distinction is important rather than whether the texts share words.

There are two families of tools to solve this need, each of which is missing. Lexical similarity methods are derived directly from the vector space model [1], [2], [3] and improved by the use of such fingerprinting methods as winnowing [5], which detect shared character or token sequences. They are fast, deterministic, and interpretable, and they form the backbone of the plagiarism detection literature [6]. Their weakness is definitional: the competency that interests the examiner most is the one they can least see, because a competent paraphrase has very few long token sequences in common with its source.

Dense representation learning is a remedy to that weakness. Distributional structure was found to have meaning [7], and the knowledge of the word representations was made usable by distributed word representations [8], [9] and the transformer encoders [10], [11], and finally, sentence-level objectives [12]-[14] yielded vectors in which cosine proximity is a good proxy for semantic relatedness. It is also a feature of contemporary embedding

services that is exposed via an HTTP interface, and principled model selection is possible with benchmarks like MTEB [15]. Meanwhile, LLMs [19] have been tasked to be the evaluators [20,21] and provide criticism that no scalar measure can.

The merger of the two is appealing and with our experience, it's surprisingly hard to pull off. The problems are not the ones highlighted in the research literature. They are operational. An embedding service might have a quota and might be unreachable; the sensible engineering solution is a fallback that can be executed locally, but a fallback that does a shoddy job of returning numbers on a different scale is a service outage that results in a wrong answer, but with full confidence. An LLM asked for JSON will generally accept it, and occasionally not, and a parser assuming it will accept it will reject it on the odd occasion when it doesn't. A system that trusts an account provides models that an inference provider publishes an endpoint for. None of these are modelling questions, and all these will help to decide the success or failure of the system.

This paper outlines a deployed system, MindMirror, based on an explicit split of the two capabilities, and presents a quantitative evaluation that regards the operational failure modes as first-class objects of study. The system does not use the LLM for the similarity score but only for the qualitative categories that a scalar cannot capture. This separation allows graceful degradation: if the LLM is not available, the similarity score is still valid, and the system advertises that.

A. Novelty of This Research Work

This research study is more about methodology than architecture. It did not introduce the combination of dense retrieval with generative models, as hybrid pipelines have been proven to be widely used and established [18]. The subject of study is new. So far, the literature is assessing such pipelines by their accuracy on curated benchmarks; here, this research study assesses one by systematically inducing, measuring, and repairing the failures that determine whether or not it works when a third-party provider misbehaves. The following 5 sets of claims of novelty are presented.

An ordinal diagnostic methodology for similarity systems. To do so, this research study builds a 5-tier ordinal corpus and demonstrates that classical summary statistics, such as Spearman correlation and ROC area, reach saturation levels on this corpus for configurations which show significantly different behavior. Thus, we introduce adjacent-tier gap analysis and the minimum adjacent gap, as a bottleneck statistic, as the discriminating criterion. This exposes ordinal-validity problems that are masked by aggregating correlations, which are relevant to any graded similarity measure.

Diagnosis and repair of a silent degradation path. We demonstrate that a fallback that satisfies all the interface contracts can fail to meet a monotonicity expectation of the similarity measure: that the measure should be monotonic; we localize the violation to unweighted term frequency; we provide the repair; and we demonstrate that monotonicity is restored. To the best of our knowledge, this beautiful degradation property of a hybrid similarity system has not been quantified before.

A negative result on local inverse document frequency. IDF (inverse document frequency) is universally useful in retrieval [2] [3]. As we demonstrate here, it is harmful to estimate it for the 2 or 3 chunks that make up a single pairwise comparison, because these chunks are the ones it ignores, and they are the evidence of similarity. This restricts the use of a basic technique.

Empirical decoupling of embedding size from task accuracy. We don't observe a monotonic trend

across 5 different models of embedding in the 384 dimensions to 1536, as the lowest-dimensional model is the best one. The dimensionality is often used as a proxy for quality, but it is not in this case.

Operational reliability as a measured property. We measure the difference between a provider's advertised model catalogue and the subset of models that an account can invoke; the per-model rate of need for post-hoc repair of structured output; the concurrency that is actually attainable from independent prompts. They are considered engineering details in the literature, and we illustrate what they represent to a user; the first is a score of 0, and the second a score of 1.

B. Contributions

- This research study defines and executes a document comparison pipeline based on two independently provided document similarity scores and qualitative analysis, enabling any document similarity score to fail without affecting the qualitative analysis, and vice versa, so that failure is detectable and reportable.
- This research study creates an ordinal evaluation corpus, which consists of 20 documents across 5 topics (30 labelled pairs) written by hand. Manual authoring of paraphrases enables the evaluation of the paraphrases without the errors of the generating system interfering with the evaluation.
- This research study quantifies the cost of the lexical fallback and demonstrates that it is not just a missing precision, but a loss of ordinal discrimination between the two bottom levels of the scale: same-topic versus unrelated document pairs; the same-or-lower level on the scale is indistinguishable from the fallback.
- This research study evaluates the structured-output performance of the language models that we have access to, and reports the percentage of recovery failures that require three levels of cascades and the percentage of disagreement between independent models on the same pairs of recoveries.
- This research study sees a difference between the advertised model and the actual model that

can be invoked, and explain the design of surfacing errors so that this class of failure is presented in the format of a diagnosis, rather than a score of zero.

The remainder of the paper is organised as follows. Section II summarizes the methods for lexical and dense similarity, the evaluation based on LLM and readability metrics with a view to the shortcomings the work addresses. In Section III, the method is formally proposed. The results of the experiment, its implications, and the limitations and threats to validity are presented in Section IV.

II. LITERATURE REVIEW

A. Lexical Similarity and Plagiarism Detection

In the vector space model, a document is described as a weighted vector over a vocabulary, and documents are compared by the cosine of the angle between them [1,3]. This combination is an excellent basic method for topical retrieval, as it reduces terms that are too common to be informative, resulting in inverse document frequency weighting [2]. Set overlap measures, such as the Jaccard coefficient [4], ignore weights entirely and only ask which terms occur together. Plagiarism detection specializes these ideas towards the exact reuse. The fingerprinting method picks a subset of the hashes of the substrings in a document, and the winnowing method is designed to do this in a way that ensures that any sufficiently long common passage is identified, while allowing index size to be kept to a reasonable bound. The job and its measures were standardized in the PAN evaluation framework [6], and a distinction was drawn between verbatim copying (which they can detect almost perfectly) and obfuscated and/or translated reuse (which they cannot).

The restriction is not merely a side-effect, but an inherent one. Any method that has evidence of shared surface sequences is blind to a rewrite that preserves propositions but changes wording; a long enough paraphrase makes long-n-gram overlap as low as unrelated text. This is also reproduced by our own measurements: Section III describes a 5-gram indicator which separates

verbatim copies perfectly and assigns an essentially the same value to heavily paraphrased pairs as to unrelated ones.

B. Dense Semantic Representations

One technique used was latent semantic analysis [7], which employed truncated singular value decomposition (trSVD) of a term-document matrix to obtain a low-dimensional space in which documents sharing no vocabulary could still be represented close together. Improving the lexical geometry was achieved by distributed word representations learned by prediction [8] or by global co-occurrence factorization [9], but the composition of word vectors into document vectors was still ad hoc.

Using static vectors instead of context-dependent vectors, as in transformer encoders [10, 11], resulted in sentence representations that were not found to be well suited for cosine comparison. Sentence-BERT [12] tackled this by learning to recognize sentence-pair similarity directly, while multi-task learning methods like the Universal Sentence Encoder [13] demonstrated that it was possible to achieve competitive sentence representations by using unsupervised objectives, like the SimCSE [14] method. MTEB [15] later systematically evaluated cross-model results on dozens of tasks.

At retrieval scale, the practical challenges are different than representation: Nearest neighbour queries over very large collections can be done via quantisation and GPU-parallel search [16] and navigable small-world graphs [17] are commonplace for retrieval; embedding pipelines are a regular part of language-model systems via retrieval-augmented generation [18]. Our deployment, which is significantly smaller than where approximate indexing is profitable, is where Section III makes its case for exact search.

C. Language Models as Evaluators

Few-shot prompting [19] opened the door to the possibility of obtaining structured judgments from language models without having to train them, and there is a large body of literature that leverages language models as judges of text. MT-Bench and

Chatbot Arena [20] showed that it is possible to achieve high levels of agreement with human preference ratings of >90%, and that certain models demonstrate position, verbosity, and self-preference biases. Chain-of-thought prompting [22] was shown to enhance the correlation with human ratings in summarization and dialogue quality by G-Eval [21].

Embedding-based automatic metrics lie in between. BERTScore [23] takes two types of word representations: the embeddings of the tokens in a specific context, and correlates them with the embeddings of the reference text. BERTScore [23] is a fixed function, not a prompted model, that correlates contextual token embeddings between candidate and reference. The difference is important with regard to system design, as a metric is going to be both easy and dependable to calculate, whereas a prompted judge will be neither easy nor dependable.

This literature does not usually include the magnitude of the machine-readability of the judgements. A model may include it in prose, fence it with Markdown, return it with a maximum number of tokens, or return single quotes and trailing commas. If you have a system that consumes a third party HTTP API, you can't necessarily impose constraints on the decoding; on the other hand, if the inference stack supports it, then you don't have that problem. In the following sections, the cascade used and the frequency of exercising each layer are defined.

D. Readability and Writing-Quality Metrics

Readability formulas estimate the reading difficulty of a text. Readability formulas use simple surface statistics to estimate reading difficulty. The Flesch Reading Ease score [25] is based on the two parameters, mean sentence length and mean syllables per word, while the Flesch-Kincaid grade level [26] rescales similar inputs to a school grade level. Gunning's fog index [27] measures grade based on the percentage of polysyllabic words, SMOG [28] uses a fixed-size sample of words to estimate grade by counting polysyllables, the Coleman-Liau index [29] changes the syllable count with character counts, and the automated

readability index [30] is similar in that it uses character counts.

These are instruments that are rightly interpreted as a measure of reading ease rather than quality. Long sentences and terms with several syllables must be used in a precise, well-organised technical argument and it will be considered to be difficult. The concrete consequence reported in Section IV is that the composite quality score that was implemented in the system under study is zero for competent academic prose, a result that is not considered as a property of the documents.

E. Gap Analysis

This work is driven by three gaps. First, lexical and dense methods are usually contrasted on accuracy; however, when a system is deployed, the question of what occurs when it silently switches one method for the other is an engineering question which the literature on accuracy does not address. We illustrate that the substitution makes it impossible to discriminate between the lower half of the scale, a more serious failure since this part of the scale is hidden in the numbers that are typically reported, and it is only the lower half that may be relevant.

Second, the focus of the work on LLM evaluators is on agreement with human judgements, and infrequently on a failure rate of the structured output, which is a critical success factor for a pipeline. Third, the literature assumes that the models that are mentioned are available. In practice, its advertised catalogue, its retired models and the collection it might call upon are three different sets, and we measure the difference between them.

III. PROPOSED RESEARCH METHOD

A. Design Principle and System Overview

The system is organized around a single principle. The system's organization is based on the only principle: the reported similarity score and the qualitative analysis must be provided by different providers to ensure that no failure can pretend to be a result of another failure. If an individual had a monolithic design, with one model producing

both, then if he failed, it would be impossible to tell that it was a failure.

MindMirror is a two-level web-based application. Upload, configuration and presentation is done in a single-page React client. The Python service hosts a REST interface with more than 24 endpoints and comprises four phases: ingestion, dense similarity scoring, language-model analysis, and report synthesis. Two external providers are used, one for embeddings and one for text generation.

Persistent state comprises user credentials, comparison history and cached document vectors.

Fig. 1 shows the pipeline. The topology of the output stage is noted: the cosine scorer is the only element that affects the calculated similarity figure, the language model and the deterministic lexical measures do not. What it does mean is the structural representation of the design principle above and it's the requirement that makes it possible for the system to say that a similarity score is valid when the generation provider has failed.

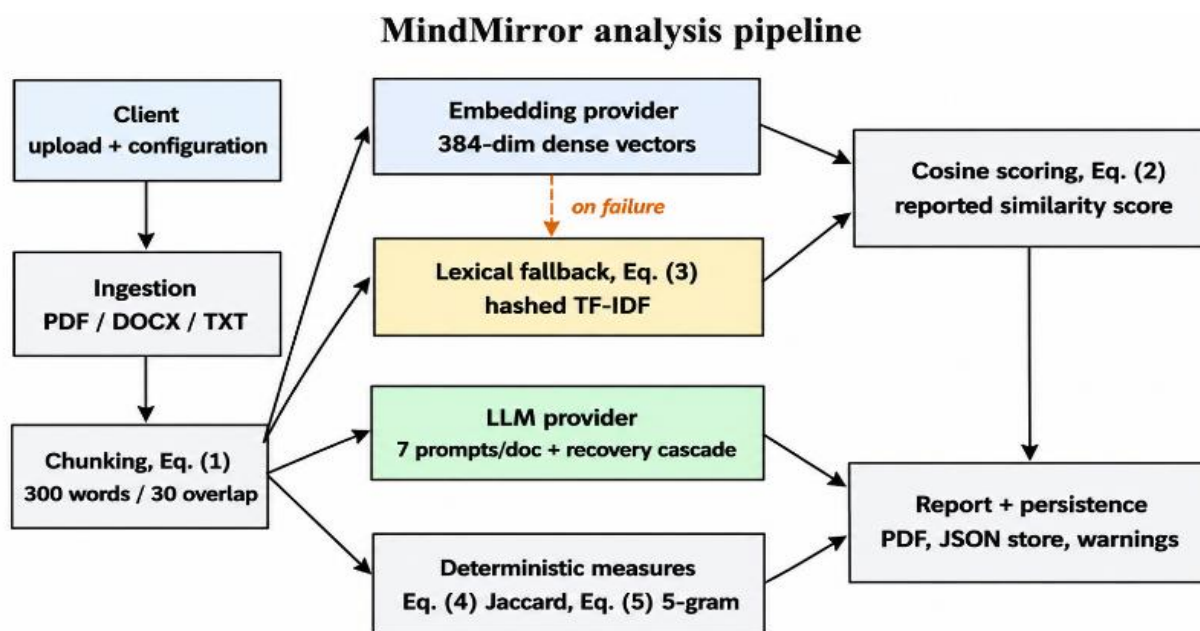


Figure 1. Proposed system pipeline

The reported similarity score derives solely from the cosine scorer; the language model and the deterministic measures feed the report. The lexical fallback (dashed) is engaged only when the embedding provider fails.

B. Design Alternatives Considered

Three options were ruled out for various reasons, some of which are pertinent to the outcomes. The first was to have the language model generate the similarity score directly, as described in the literature on the LLM-as-judge [20], [21]. We dismissed it because otherwise the reported number would have picked up the model's run-to-run variance and its availability, which we

quantified in Section IV-G, showing 5.91 percentage points disagreement across models on the same inputs.

The second one was to be the lexical fallback itself was removed, and to fail closed if the embedding provider is not available. This is defensible and indeed better than what was built; we kept the fallback as part of the code because a comparison that returns nothing is bad user experience, and during the evaluation we found that the fallback's output is not comparable to the primary path. The compromise is the disclosure mechanism of Section III-L; the argument for failure closed is made by Section IV-E.

The third was to take a hybrid approach and combine the lexical and dense scores into a single weighted score as in hybrid retrieval. We rejected this because the two signals address different questions; reuse of words and similarity of content; and a mean of the two addresses neither question. Both are thus reported separately, and in Section IV-D, they are seen to be complementary and not redundant.

C. Document Ingestion and Normalisation

Suppose d is a document that has been uploaded in PDF, DOCX or plain text format. Extraction

D. Chunking and Dense Representation

Normalized text is broken down into overlapping word windows. The length of the windows $L = 300$, overlap $o = 30$, and stride $L - o = 270$, thus for a document of n words, the chunk set is

$$C(d) = \{w, [i:i+L], : i = 0 \ 270 \ 540 \ \dots \ i < n\} \quad (1)$$

The overlap ensures that a passage that crosses a window boundary is entirely within one of the windows so that there is no suppression due to window boundary effects. The embedding function $e: \text{chunk} \rightarrow \mathbb{R}^D$ is a function that takes a chunk as an argument and returns a unit-normalised vector in $\mathbb{R}^D$, with $D = 384$. Requests

returns a sequence of characters, while normalisation merges groups of whitespace, converts multiple line separators to a single one and removes null characters. Plain text input is attempted to be UTF-8 decoded, then Latin-1 and UTF-16, if that fails. An extraction mode may optionally keep only tabular lines (which are indicated by multiple tab characters) or pipe delimiters or a numeric-token pattern, or it may exclude tabular lines.

are batched and rate limit responses are retried with exponential backoff.

E. Similarity Computation

Let $R = C(r)$ be the reference chunks and $Q = C(q)$ the comparison chunks. For each comparison chunk we take its best match against any reference chunk, and average those maxima:

$$s(q, r) = \frac{100}{|Q|} \sum_i \max_j \cos(e(q_i), e(r_j)) \quad (2)$$

with $\cos(u, v) = \langle u, v \rangle / (\|u\| \|v\| + \epsilon)$ and $\epsilon = 10^{-9}$ guarding against a zero vector. The max-pooling over the reference is intentional: a comparison passage that restates any part of the reference should obtain a high score even if the reference is lengthy and primarily is about different topics. Averaging over Q will give weight to documents with a high proportion of content matching, meaning that a short verbatim quote contained within a longer original document will not obscure the score. The result is clipped to $[0, 100]$. Both variables are unit-normalized, so that cosine becomes an inner product, and the entire computation for each pair becomes just one matrix-vector product for each comparison chunk.

The scoring is done exactly using a dense linear-algebra library, and the exact search can be done in microseconds; the vectors are thus represented as JSON and the system performs approximate indexing, if required, but it is not done at the corpus sizes this system supports.

F. A Family of Chunk-Aggregation Functions

One family member is given in equation (2). The aggregation step is any function that maps $|Q| \times |R|$ cosines to a scalar, where $|Q|$ and $|R|$ are the number of examples in a query and a single response, respectively, and $e(q_i)$, $e(r_j)$ denote the examples in the query and the response,

respectively. Five options are considered in Section IV-D:

$$A_{\text{mean}\backslash\text{mbox-max}}(S) = \frac{1}{|Q|} \sum_i \max_j S_{ij} \quad (3a)$$

$$A_{\text{max}\backslash\text{mbox-max}}(S) = \max_{i,j} S_{ij} \quad (3b)$$

$$A_{\text{mean}\backslash\text{mbox-mean}}(S) = \frac{1}{|Q| |R|} \sum_i \sum_j S_{ij} \quad (3c)$$

$$A_{\text{sym}}(S) = \frac{1}{2} \left[\frac{1}{|Q|} \sum_i \max_j S_{ij} + \frac{1}{|R|} \sum_j \max_i S_{ij} \right] \quad (3d)$$

The default equation (3a) and its most recall oriented cousin (3b) which reports the single best matching passage in any location. The weightings given in each of the equations are equal, and (3d) symmetrises (3a) so that a short comparison document scores as it would in the reverse

equation. There are two more that are less common. Optimal assignment forbids several comparison chunks from all claiming the same reference chunk, by maximizing over injective matchings M :

$$A_{\text{assign}}(S) = \frac{1}{n} \max_M \sum_{(i,j) \in M} S_{ij}, \quad n = \min(|Q|, |R|) \quad (3e)$$

Note that this is a lower bound on the optimal solution in the Hungarian problem (the greedy solution is sufficient for the number of chunks considered). Finally a temperature-weighted soft

maximum interpolates between (3a) and (3c), allowing several good partial matches to contribute, but not the best.

$$A_{\text{soft}}(S) = \frac{1}{|Q|} \sum_i \sum_j \sigma_{ij} S_{ij}, \quad \sigma_{ij} = \text{softmax}_j \left(\frac{S_{ij}}{\tau} \right) \quad (3f)$$

with $\tau = 0.05$. As $\tau \rightarrow 0$ this recovers (3a) and as $\tau \rightarrow \infty$ it recovers (3c). In Section IV-D it is seen that the choice between these is irrelevant for single-chunk documents and crucial for multi-chunk documents.

G. Lexical Fallback

If the embedding service is not available, not authorized or out of quota, the system uses a local lexical embedding to complete a comparison. Each token is mapped to a fixed coordinate using a hashing trick [36] and the frequencies will be stored in the vector.

$$v_{h(w)_D} = \text{tf}(w), \quad v \leftarrow \frac{v}{\|v\|_2} \quad (4)$$

The number of times the token w occurs divided by the total number of tokens, which is denoted by $tf(w)$, and where h is an MD5 digest of that number interpreted as an integer. The output dimension is the same as the dense path; the same scoring code is used without changes. By construction the substitution is clear to the pipeline; in fact, it is this property that makes it dangerous, because it produces numbers that cannot be compared with those that the dense path produces.

$$v_{h(w)D} = 1 + \log (tf(w)), \forall w \in T(x) \setminus W, v \leftarrow \frac{v}{\|v\|_2} \quad (5)$$

The third correction is the inverse document frequency, which is in the literature cited by the retrieval experts as a must-have correction [2, 3]. Since a pairwise comparison does not have any corpus, the document frequency can only be

$$idf(w) = \log \left(\frac{N + 1}{df(w) + \frac{1}{2}} \right) \quad (6)$$

Section IV-E reports that (6) is counterproductive and provides a rationale for that; for N of order two or three, the words that appear in both documents are all the evidence of similarity, and IDF discounts those hardest. Therefore, (5) applies without (6), which is what we discover when we measure it.

$$\min_g \sum_k (g(x_k) - y_k)^2 \quad (7)$$

Note that x_k and y_k are the lexical and dense scores in pair k . The assumption of monotonicity, property (5), is the only property we are ready to assert, and therefore we prefer isotonic regression to a parametric fit. The map is therefore left out of the evaluation, so it is evaluated with the data left out of the map, which is called leave-one-out in Section IV-E.

I. Multi-Dimensional Language-Model Analysis

Each of the seven prompts returns an independent set of JSON objects, including a semantic

In Section IV-E, it is concluded that this formulation is ordinaly invalid and that the reason for this is that high-frequency function words which appear in similar proportions in any two English prose documents tend to dominate the vector, irrespective of subject. Two normal corrections are thus added. Let $T(x)$ be the multiset of tokens of x and W be a closed class of stopwords. Repaired embedding, stops words and reduces repeating words:

approximated within the chunks of the two documents being compared, $N = |C(r)| + |C(q)|$, where terms that occur in both documents are dampened with smoothing instead of being eliminated:

H. Calibrating the Fallback onto the Dense Scale

The ordering is needed but not enough; it is a dense-path score on an incomparable ordering that leads the reader astray. Therefore, we fit a monotone mapping between the lexical and dense scores using isotonic regression [37, 38] – the solution that is “pool adjacent violators”.

similarity object containing a set of similarities and differences, a content nature object containing type, formality, purpose, audience, domain and structure, an effectiveness assessment object containing coverage, accuracy and completeness sub-scores, a set of prioritised improvement recommendations, and a tone object containing emotional valence, intensity and objectivity. Two additional categories (readability and originality) are calculated deterministically, as described below.

The temperature for classification and scoring categories is set to 0.1, and for generating recommendations, to 0.3, the latter being suitable when multiple suggestions are desired. Each prompt contains a literal example of a JSON structure that matches the schema described, and we discovered that this is more effective than a

natural language schema description when getting the output correct in terms of being parseable.

The five language-model categories do not share any state, and are issued concurrently, not sequentially. The sequential wall-clock cost is the sum and the concurrent cost the maximum of all the writing t^k of all the categories k :

$$T_{\text{seq}} = \sum_k t^k, T_{\text{con}} = \max_k t^k \quad (8)$$

For similar calls, Equation (8) would predict a five-fold speedup. The measured figure is much lower, reported in section IV-J, and they attribute this to provider-side serialisation. Concurrency is still a worthy approach because it's the way that makes a single-document comparison fit into the concurrency limit of a serverless platform.

J. Structured-Output Recovery Cascade

The generation provider is used through HTTP, and as there is no constraint in decoding the response, its content cannot be assumed to be well-formed. A cascade recovers a recoverable object (if one exists).

1) Strict parse

JSON is directly being parsed as the answer. This works with only the object being emitted from the model.

2) Repair

Markdown fences are stripped, the substring between the first opening and last closing brace is

extracted, control characters are removed, and a sequence of targeted rewrites repairs the failure modes observed in practice: trailing commas before a closing delimiter, single-quoted strings, and unquoted keys.

3) Field salvage

If the structure fails to be repaired, regular expressions will recover individual fields: a percentage after a score like keyword, bulleted list items, and tone keywords provide a partial object, not none.

There are typed defaults for each category, and if the category fails completely, it returns a well-formed response with an explicit error field, instead of a missing key. The frequency of use of each layer is reported in Section IV.

K. Deterministic Originality Measures

The dense score is complemented by two lexical measures. Writing $W(x)$ for the token set of document x , originality is the Jaccard complement

$$O(q, r) = 100 \cdot \left(1 - \frac{|W(r) \cap W(q)|}{|W(r) \cup W(q)|} \right) \quad (9)$$

whereas the plagiarism indicator for x and $G_5(x)$ is the percentage of the five-grams in the comparison document that are also found in the reference:

$$P(q, r) = 100 \cdot \frac{|G_5(r) \cap G_5(q)|}{\max(1, |G_5(q)|)} \quad (10)$$

The asymmetric denominator is deliberate because it is not the fraction of the source that is reused, it is the fraction of the submitted document that is reproduced. Once P falls below 30%, it falls into

bands of O , which are labeled as O once thresholded at 15%.

L. Failure Disclosure

The two providers have different schemas for the response, since they fail independently. These are

aggregated, normalised (removing per-request correlation identifiers), deduplicated, and returned along with the results in a warnings array. Errors by the provider are mapped to human-readable text, differentiating between the unauthorised model, retired model, invalid credential, and rate limit, and the client displays an unauthorised model, retired model, invalid credential, or rate limit banner, respectively, while the similarity score is still valid, and the qualitative sections are not. This mechanism does exist because the untranslated behaviour returned a score of zero, which is not different from a true finding of no similarity whatsoever.

M. Report Synthesis and Persistence

The results are displayed in the order of the descending similarity and output to a paginated PDF document that includes a cover page with document metadata and a ranking table, with one page per comparison document for each of the seven analysis categories. Interruption of a write does not destroy state: comparison records are written to a write-then-rename, state-preserving JSON store, and monotonic identifier allocation means that deleted records are moved to a recoverable trash instead of being deleted.

N. Implementation and Reproducibility

The service is deployed using Python 3.12.3, an ASGI web framework, and the client, using React and a module-bundler development server, a proxy for API traffic. Embed-english-light-v3.0 is the model that is used for embeddings ($D = 384$).

Dependencies are version-pinned. The evaluation harness in Section IV calls the deployed service objects directly, instead of reimplementing them, thus the reported numbers apply to the shipped system; corpus, harness and generated artifacts are provided with the source.

IV. RESULTS AND DISCUSSION

A. Experimental Setup

The evaluation corpus consists of 20 hand-written documents on 5 topics (machine learning, Baroque music, ocean acidification, judicial review and mitochondrial biology), with an average of 130.4 words (range 109-149). There are four variants of each topic—one verbatim copy, one light paraphrase (that is, one that restores the propositions and order), one heavy paraphrase (one that restores the facts but changes the structure), and one same-topic document that is a different document. There are tiers from 4 to 1, with tier 4 being the easiest and tier 1 being the most difficult. Tier 4 is the easiest because it is pairing a reference with its variants; tier 1 is the most difficult, because it is pairing a reference across topics. This results in 30 pairs marked, as summarized in Table 1.

No machine was used for paraphrasing. If the labels had been generated by machine paraphrase then any conclusions about the system under test would have been confounded by the behaviour of the generating model. All documents do not exceed the 2000 character truncation limit and truncation is consistent between tiers.

Table 1. Composition Of The Ordinal Evaluation Corpus.

Tier	Label	Pairs
4	Identical (verbatim copy)	5
3	Light paraphrase	5
2	Heavy paraphrase	5
1	Same topic, different content	5
0	Unrelated topics	10

B. Construct Validity of the Similarity Score

The similarity score by tier for both embedding backends is provided in Table 2. The ranking correlation to the ground-truth ordering is $\rho =$

0.9745 and the tier is monotonically decreasing from 100.00% (verbatim copies) to 11.36% (unrelated pairs) with dense embeddings. The separation between the most extreme tiers is 88.64

percentage points, Cohen's $d = 11.92$ for the pairs that were copied or lightly paraphrased compared

with unrelated pairs, which is well beyond the standard threshold for a large effect [32].

Table 2. Similarity Score By Ground-Truth Tier, Mean $\pm$ Standard Deviation, For The Dense-Embedding Path And The Lexical Fallback. Source: Results/Metrics.Json, Similarity.

Tier	n	Dense (%)	Lexical (%)
Identical	5	100.00 $\pm$ 0.00	100.00 $\pm$ 0.00
Light paraphrase	5	95.08 $\pm$ 2.03	90.35 $\pm$ 2.31
Heavy paraphrase	5	68.06 $\pm$ 7.79	65.97 $\pm$ 4.76
Same topic	5	45.74 $\pm$ 4.23	57.00 $\pm$ 8.53
Unrelated	10	11.36 $\pm$ 9.80	57.60 $\pm$ 5.93

When tiers 4-2 are considered positive and tiers 1-0 are considered negative (that is, if it is possible to recognize a document that has been heavily rewritten from the reference), a dense path can get ROC AUC = 1.0000 [33]. The best value of Youden's J [34] is 60.24% with precision and recall of 1.00 and 1.00 respectively. A perfect separation on the basis of such a corpus should at this stage

be understood as evidence that the tiers are well separated, rather than as an indication of infallibility, and this is developed further in Section IV-O. In each of the storyboards, the plot of each pair is displayed, and the tiers are seen to be not only different in mean but also disjoint under the dense path.

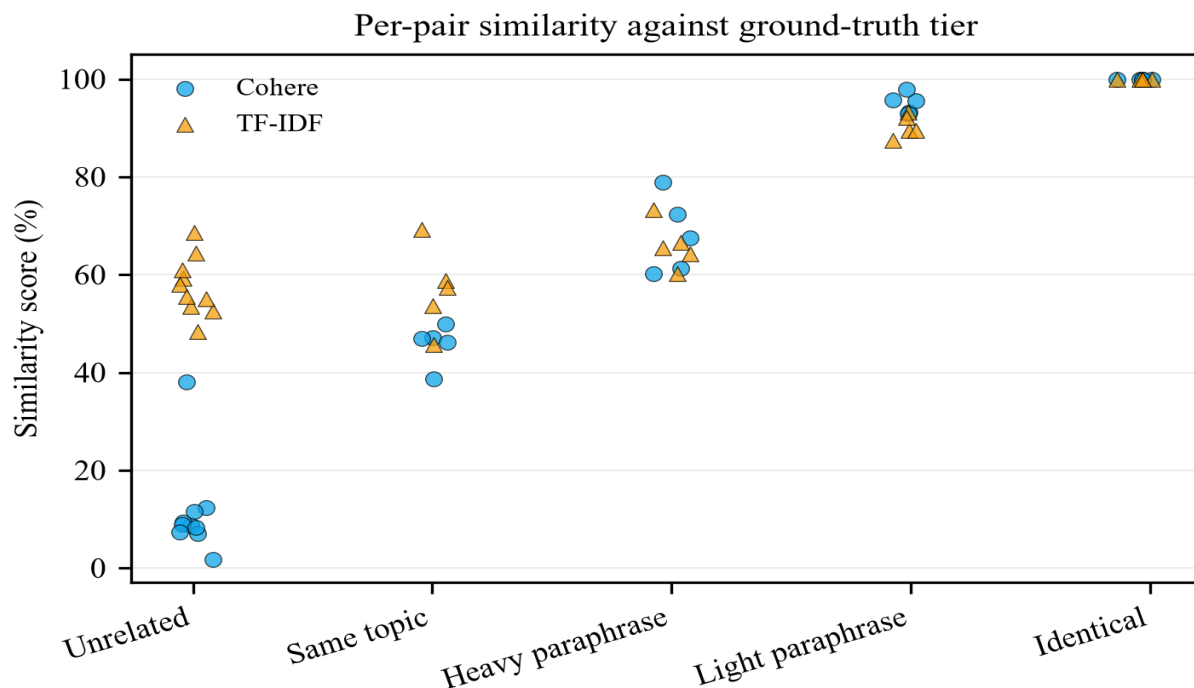


FIGURE 2. Per-pair similarity against ground-truth tier for both backends. Horizontal jitter is added for legibility. The lexical fallback compresses tiers 1 and 0 onto indistinguishable values.

Fig. 3 gives the corresponding ROC curves. The dense path achieves the unit true positive rate

when the false positive value is zero, and increases in steps until it reaches saturation, at which point

the documents classified as same-topic are considered to be reference-derived. The region between the two curves is concentrated in the low false-positive region; in the region that a screening

tool would actually operate, a reviewer who is tolerant of few false accusations is getting a lot more for his or her buck from dense representations.

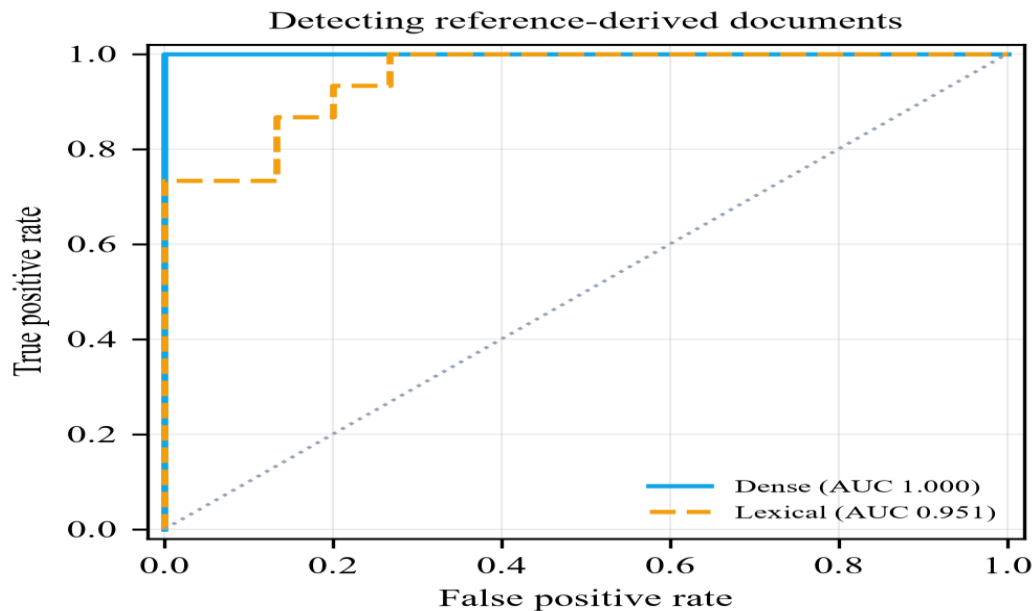


FIGURE 3. Receiver operating characteristic for discriminating reference-derived documents (tiers 4–2) from independent ones (tiers 1–0). The dotted diagonal is chance.

It's worth to be clear about what is being provided by tier 1 here. Given a similarity measure, the case that the adversarial document must be distinguished from reuse is a same-topic document; students who are given the same assignment write close, topically similar documents, independently. That the dense path scores these 45.74%, and scores heavy paraphrase 68.06%, shows that it separates out the content of the shared subject matter, from the content of the shared substance, which is the property the application needs.

C. Embedding Representation and Task Framing

All 5 embedding models on the free tier recover the ordering of the ground-truth ordering, and all of them have the same Spearman correlation (0.9745) and the same area under the ROC curve (perfect score). This is not accidental, but a feature of the measurement itself: For any measurement that can be represented by a five-level ordinal scale, and where many ties are present, the ranking of the five levels can be done optimally for the maximum attainable rank correlation. Only reporting such a statistic would have masked actual differences between configuration and that is why the adjacent-gap analysis that is introduced in Section 1-A is reported.

TABLE 3. Embedding models ordered by narrowest adjacent-tier gap. All are monotone; all attain Spearman 0.9745 and ROC AUC 1.000. Source: results/metrics.json, v2.e7.

Model	Dim	Min gap	T2-T1	T1 mean	T0 mean
multilingual-lightv3.0	384	5.92	21.89	48.01	16.18
v4.0	1536	5.48	24.82	34.17	17.23

english-light-v3.0	384	3.90	17.24	57.38	24.42
english-v3.0	1024	2.93	19.13	56.32	30.77
multilingual-v3.0	1024	1.40	18.33	63.70	38.61

On the bottleneck criterion the leader is multilingual-light-v3.0 with 5.92 points while the laggard is multilingual-v3.0 with 1.40 points. The first is 384-dimensional, the last is 1024-dimensional; the best-separating model is quite a bit smaller than the ones available, the worst is

several times larger. As seen in Fig. 4, there is no monotone relation between dimensionality and separation across the 5 models, with the right panel displaying the relation between separation and dimensionality.

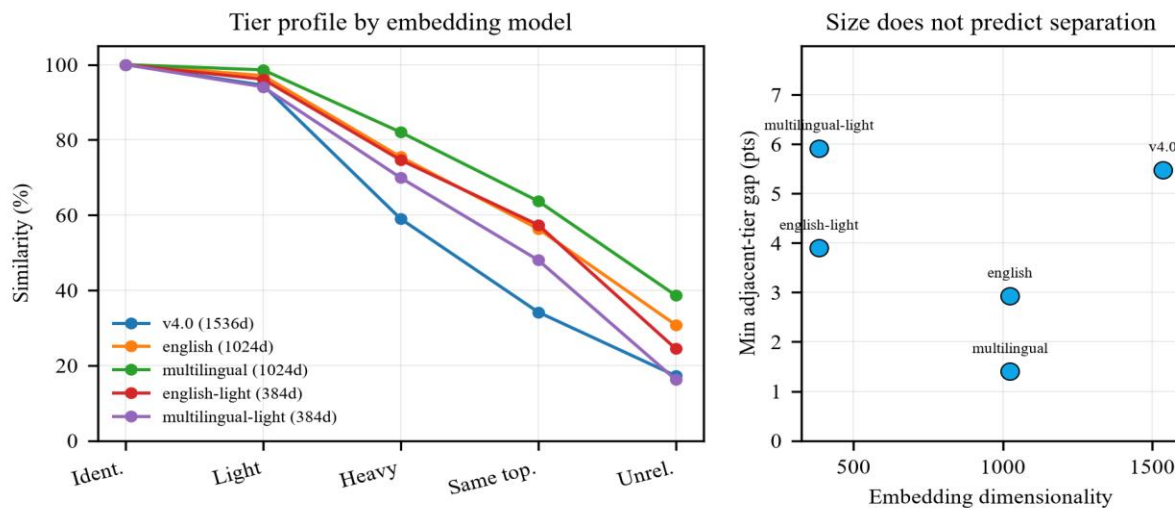


FIGURE 4. Left: tier profile for each embedding model. Right: dimensionality against the narrowest adjacent-tier gap; the relationship is not monotone.

This is a strong claim and we're very intentional about it. The difference between the two top models is less than one point, which doesn't exceed the noise level in 30 pairs of observations, so we don't state that one named model is better. The negative claim is the one that is useful, and the data back it up: dimensionality is not seen to

improve the accuracy of the task, and enlarging it comes at a higher proportionate cost in storage and bandwidth. If a practitioner used the price of a product as a proxy for its size (a heuristic that is often used), he would choose the more expensive and less accurate choice.

TABLE 4. Task-framing ablation. Three of the four accepted input_type values are indistinguishable; the asymmetric retrieval framing is marginally worst. Source: results/metrics.json, v2.e8.

Input_type	Min gap	T2-T1	T0 mean
search_query	5.92	21.89	16.18
classification	5.92	21.89	16.18
clustering	5.92	21.89	16.18
search_document	5.25	22.90	7.94

There is a need for an `input_type` that indicates task in the downstream. Document comparison is symmetric, but it was originally made to be the corpus side of an asymmetric retrieval pair. It is principled to correct this, and the measurement indicates a small effect: the asymmetric framing resulted in a minimum of 5.25 points whereas the symmetric framing, which are identical to each other, resulted in a minimum of 5.92 points. This is a negative outcome, as per Table 4, this parameter is not particularly significant, and getting it wrong is not costly.

D. Chunk Aggregation and Document Length

On the primary corpus the six aggregation functions of (3a)-(3f) produce identical results to four decimal places. It is a structural reason; namely, at average 130.4 words per document, every document is in a single 300-word window, meaning $|Q| = |R| = 1$ and each member of the

family becomes a single scalar cosine. That is a degenerate situation, not a finding; and is a corpus limitation that the first evaluation never came up.

We therefore constructed a second sub-corpus where passages from other topics were added to each document; 30 pairs with 2 to 3 chunks per document. The extension material is distributed between the reference and the comparison document of each pair, and this is by design: the documents that are actually used in the real world share boilerplate, which includes assignment briefs, journal templates, standard front matter, etc., and a similarity measure must be able to deal with this boilerplate. Thus, the sub-corpus is not just length, but aggregation under shared boilerplate, and that is what we describe it as, and not a clean length manipulation. The results are presented in Table 5 and Fig. 5.

TABLE 5. Aggregation on the multi-chunk sub-corpus with shared boilerplate. Source: results/metrics.json, v2.e13.

Aggregation	rho	AUC	T4 mean	T1-T0
mean of mean	0.8900	0.9556	65.70	12.32
softmax pool	0.8707	0.9689	100.00	-0.10
mean of max	0.8668	0.9644	100.00	-0.24
symmetric mean of max	0.8668	0.9644	100.00	-0.24
optimal assignment	0.8668	0.9644	100.00	-0.24
max of max	0.5479	0.6978	100.00	0.15

The differences are now great. Max-of-max (3b) has been reduced to a Spearman of 0.5479 by shared boilerplate: the best match of any pair is the only one that contains any information about the actual relationship between the documents, and all of the tiers score well above 97.92%. The deployed default Mean-of-max (3a) has a lower degradation path to 0.8668, but this is still not a discrimination that matters, as same-topic and

unrelated pairs are separated by -0.24 points, or none.

The bottom of the scale (12.32 points) is the strongest for Mean-of-mean (3c) with a rating of 0.8900, the only one that still has usable separation. It pays by giving away a property that are relying on: Identical documents score 65.70%, not 100%. Average over all chunk pairings lowers even a perfect match. There is no family member that is both boilerplate strong and top-notch.

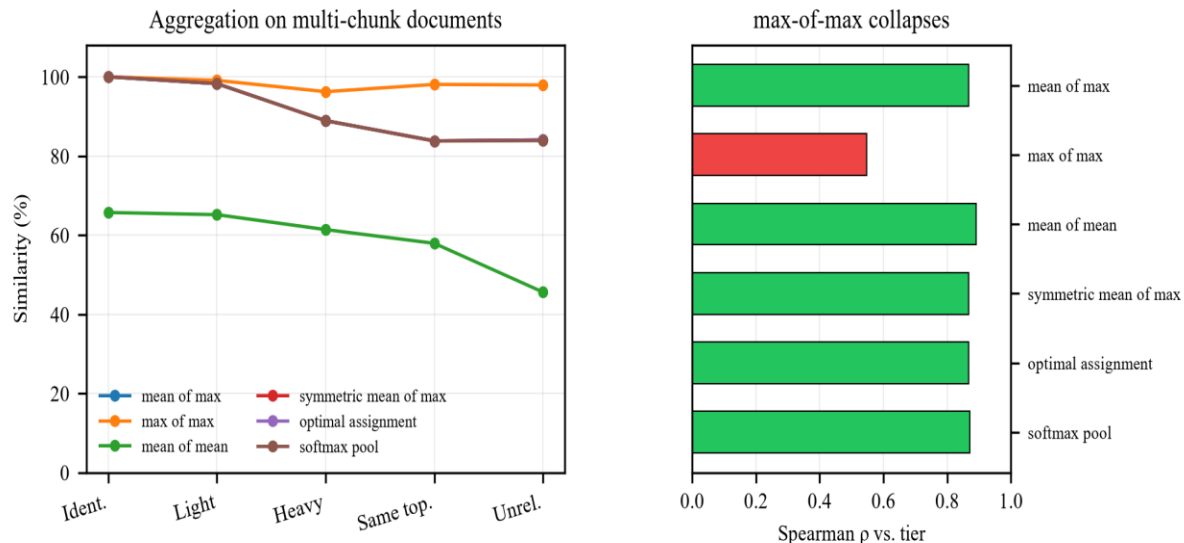


FIGURE 5. Aggregation on multi-chunk documents with shared boilerplate. Left: tier profiles. Right: rank correlation, with max-of-max collapsing.

We consider this an open instead of a solved problem and state it as such. The quality of the deployed system is retained in (3a) as it is the one that users will check first, and documents the boilerplate vulnerability, rather than hiding it. The next subsection justifies considering chunks with many references with less weight, a principled fix.

E. Diagnosing and Repairing the Lexical Fallback

The fall back (4) is invalid in an ordinal sense and adjacent-gap analysis is able to localise the failure exactly. Its closest neighbouring gap is -0.24 points, which is negative indicating the tier means are not even weakly ordered, but its Spearman correlation is 0.8415 which is still high enough to consider it as acceptable. If only the correlation is reported, then this configuration would have passed, which is the methodological aspect of Section I-A, and it is plotted in Table 6.

TABLE 6. Incremental repair of the lexical fallback. Source: results/metrics.json, v2.e10.

Lexical variant	rho	AUC	Min gap	Monotone
raw TF, Eq. (4)	0.8415	0.9556	-0.24	NO
+ stopword removal	0.9574	1.0000	6.33	yes
+ sublinear TF, Eq. (5)	0.9550	0.9956	6.63	yes
+ local IDF, Eq. (6)	0.7028	0.7867	-1.63	NO
plus sublinear tf + isotonic	0.9344	0.9822	6.05	yes

The negative effect of suppressing the closed-class function words is enough on its own to return monotonicity: a narrowest gap of -0.24 becomes 6.33 points, a correlation of 0.8415 becomes 0.9574, and an area under the curve of 1.0000. The inclusion of sublinear term frequency does

not make much difference (6.63 points); we take it for granted because of the conventional wisdom that a term that has a frequency of 20 should not overpower 20 different topical terms, but we make no specific claim that it is a requirement of the measure.

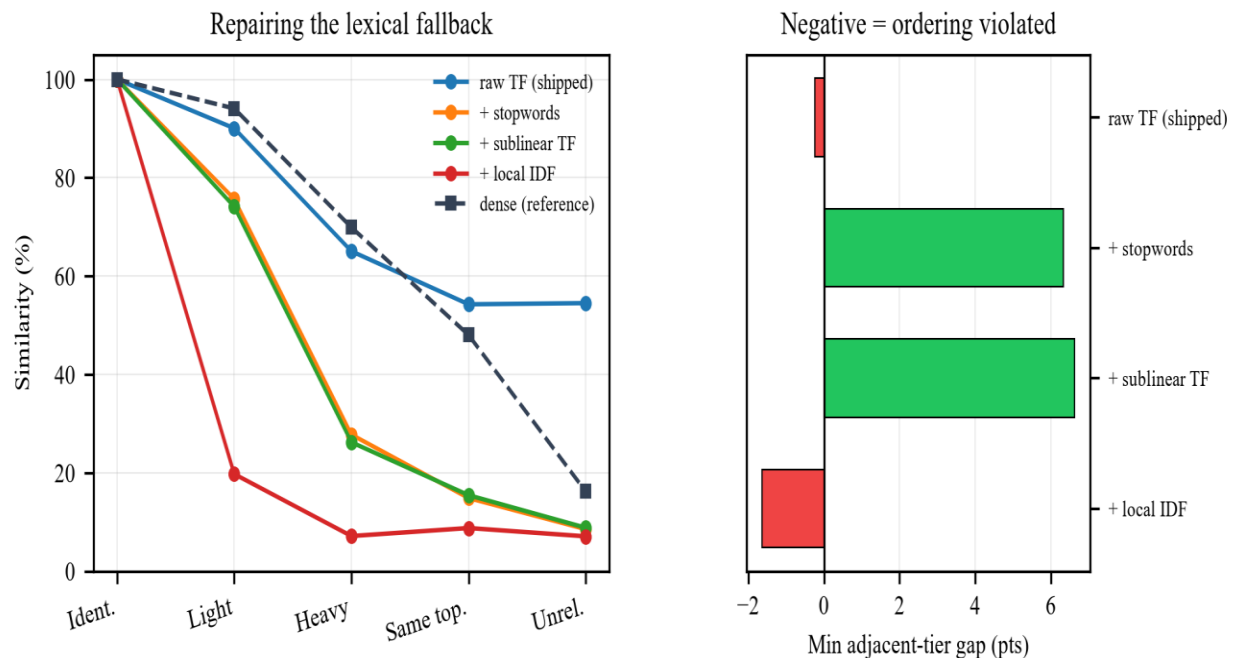


FIGURE 6. Left: tier profile of each lexical variant with the dense path for reference. Right: narrowest adjacent-tier gap; negative values indicate a violated ordering.

This is the inverse-document-frequency result; the one we did not expect. Lexical retrieval, which is close to universal in IDF weighting, is used in [2] and [3] and is applied in the standard way in (6). In the meantime it is actively harmful here: the narrowest gap turns to -1.63, monotonicity is lost again, the correlation becomes 0.7028 and the area under the curve 0.7867. The most apparent is that near-verbatim copies of documents (lightly paraphrased) receive only 19.73% while those without IDF receive 74.10%.

The explanation is based on the construction. The document frequency in (6) is estimated over N of the order two or three, as a pairwise comparison provides no larger set. A term which occurs in both of them will have a df of close to N and will be given the minimum weight, but in a pairwise comparison the terms common to both documents will be the whole evidence of similarity. IDF ignores terms that are not discriminative between documents in a corpus,

and applied to a two-document corpus discounts just the signal being measured. The generalisable lesson is that IDF assumes a collection that is big enough for document frequency to estimate something; and a pairwise comparison is not a big enough collection.

Finally, we want to ask if it is possible to get the repaired lexical score to be comparable to the dense score and not just correctly ordered. The average absolute error for the fitting of (7) with more than 30 pairs is 9.44 points with an average rank correlation of 0.9344 with the dense score. There is nothing to be gained from calibration, and the minor decrease from 0.9550 is attributed to the tie-merging in the pooling step – which is why the map is monotone, from the start – and is not a factor to consider. What it offers is scale compatibility: a disclosure obligation becomes truly graceful degradation – a fallback score is on the same scale as a dense one, within about 9 points, as Fig. 7 illustrates.

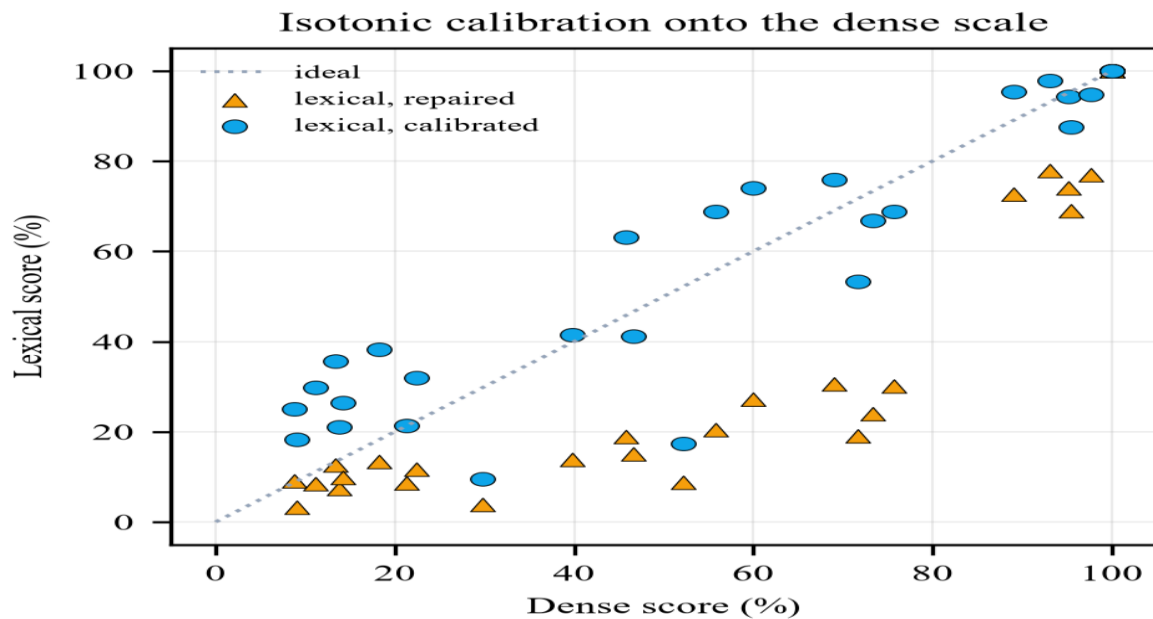


FIGURE 7. Repaired lexical score against dense score, before and after isotonic calibration. The dotted line is equality.

We DO NOT enable calibration by default. The map is created with 30 pairs from one corpus, and a monotone transform created from such small amount of data might not transfer across different documents of different registers and length. If the mechanism is reported, with an estimated error, then the deployed system will reveal the degradation, which is more defensible at this sample size.

C. Embedding Ablation: The Cost of the Fallback

The main quantitative result of the paper is the ablation, which is worse qualitatively than a simple loss of accuracy. At the highest end of the scale, both backends perform well: verbatim copies achieve 100% accuracy for both, heavy paraphrase is distinguished from light paraphrase by both. The difference is at the lowest point. In the dense path, the same-topic pairs have a higher percentage (45.74%) than the unrelateds (11.36%), which is clearly ordered. The lexical fallback gives them 57.00% and 57.60% respectively, which are indistinguishable.

There is no demonstrated reversal of discrimination and the defensible claim here is loss

of discrimination. The two lowest tiers are 0.60 percentage point apart with standard deviations 8.53 and 5.93, respectively, and are inverted in this sample, but the difference is well within the noise and we don't claim a significant inversion here. The data do confirm the lack of usable information between a document with the same topic and one with another topic in the fallback, while the dense path is 34.38 points apart. Likewise, the rank correlation decreases from 0.9745 to 0.8358, and the separation between the extremes decreases from 88.64 to 42.40 points.

The mechanism is very simple. The raw term frequency is used for the hashing embedding of (4) so that high-frequency function words that any two documents of English prose would contain in similar proportion dominate the vector. Two documents with similar registers, irrespective of subject, thus have nearly parallel vectors. The low end of the scale collapses because the absence of IDF weighting, which is known since the earliest work on term weighting, to be essential [2]. The comparison is summarised in Table 7 and the tier means are plotted side by side in Fig. 8.

TABLE 7. Ablation summary. A negative same-topic minus unrelated difference indicates ordinal inversion. Source: results/metrics.json, similarity.

Measure	Dense	Lexical
Spearman ρ vs. tier	0.9745	0.8358
Pearson r vs. tier	0.9667	0.9068
Tier 4 – tier 0 (pts)	88.64	42.40
Cohen's d (derived vs. unrel.)	11.92	6.67
ROC AUC (derivation det.)	1.0000	0.9511
Same-topic – unrelated (pts)	34.38	-0.60

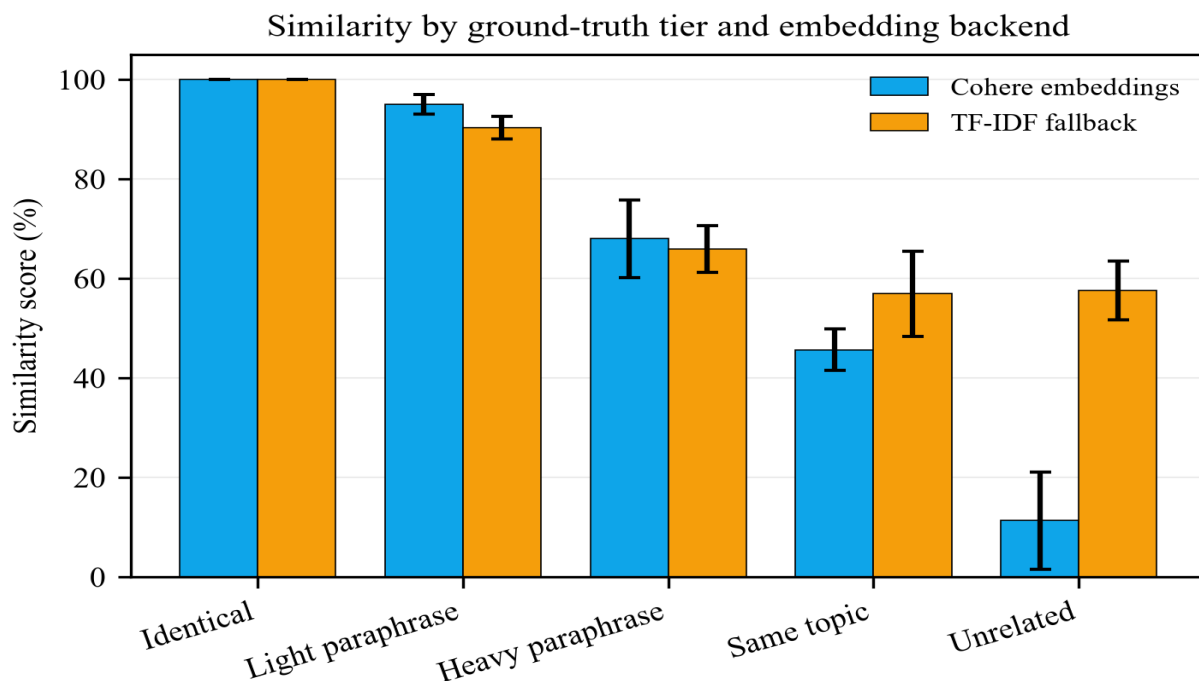


FIGURE 8. Mean similarity by tier and backend; error bars are one standard deviation. The lexical fallback fails to separate the two lowest tiers.

The engineering decision here is that the return of a fallback on a non-identifiable scale should not be silent. The same presentation as a dense-path score by the fact that it reports 57.60% similarity between documents on unrelated subjects is a correctness defect and not a degradation of quality – a user cannot tell it from a true finding. The system then makes the active backend available to settings, and reports on the difference.

D. Deterministic Originality Measures

The plagiarism indicator (5-gram) and the Jaccard originality score by tier for the five-gram plagiarism indicator are listed in table 8. There is perfect separation of verbatim reuse, as the ROC AUC = 1.0000 for the tier-4 against all-other-tiers discrimination, and the behaviour after it is as expected from the limitation described in Section II-A

TABLE 8. Five-gram plagiarism indicator and Jaccard originality by tier. Source: results/metrics.json, originality.

Tier	Plagiarism (%)	Originality (%)
Identical	100.00	0.00
Light paraphrase	20.32	42.05
Heavy paraphrase	0.00	86.69
Same topic	0.00	91.16
Unrelated	0.00	93.38

A score of 0.00% is achieved by heavily paraphrased pairs while unrelated pairs achieve a score of 0.00%. The measure, then, conveys virtually no information regarding the distinction that is most important in the academic integrity enterprise, namely whether some piece of material is the product of an independent treatment of the subject or a conscious rewriting of a particular

source. This is not a bug in (10), but rather a feature: the expectation is that evidence is made up of common sequences of 5 words, and it is for this reason that the dense path fills the gap, due to the fact that the dense similarity treats the same heavily paraphrased pairs as 68.06%. The overall distributions are displayed in Fig. 9.

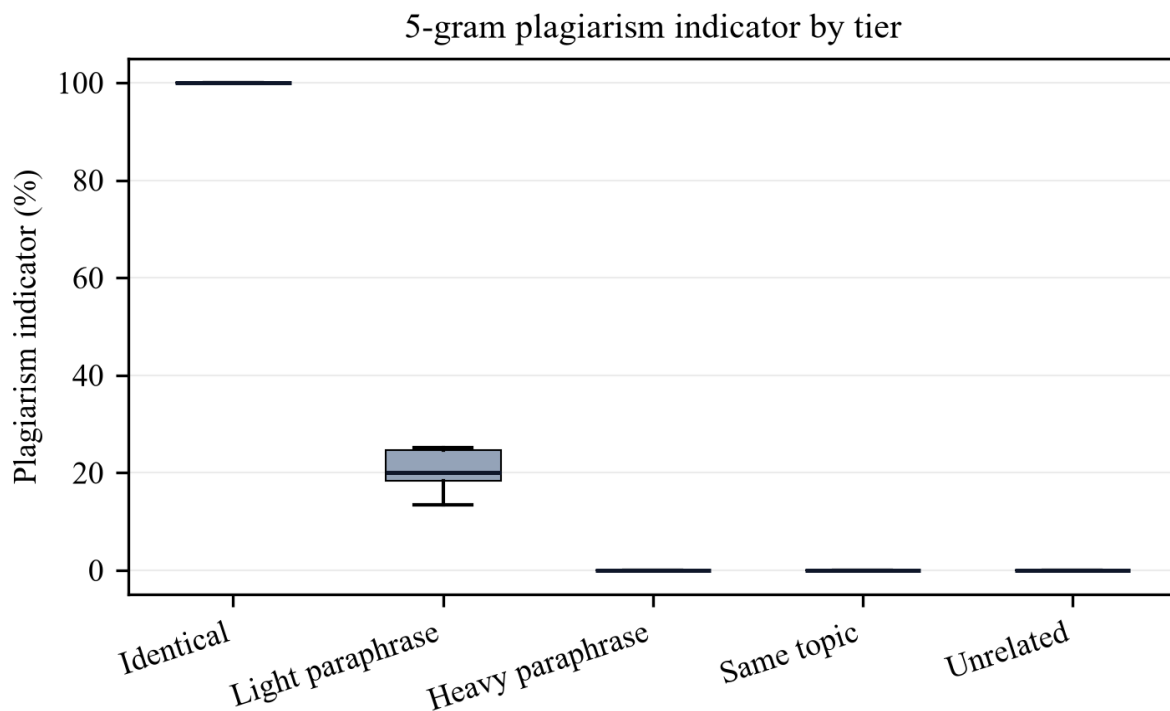


FIGURE 9. Distribution of the five-gram plagiarism indicator by tier. Discrimination is confined to verbatim reuse.

The two measures are thus complementary and both are reported by the system. The lexical indicator is a high precision evidence of copying, trivially explicable by the user (the shared texts can

be shown), the dense score is a lexical measure that lacks this precision but gains in interpretability.

E. Language-Model Agreement and Structured Output

Structured-output behaviour was checked across 80 generation requests across all the models offered to the test account. Sixty-one (76.2%) of

these were found to be able to be parsed as JSON the first time, 18 (22.5%) were able to be structurally repaired and then parsed as JSON, while 1 (1.2%) were never recovered. This is broken down by model in Table 9.

TABLE 9. Per-model structured-output compliance, usable-response rate, rank correlation against tier, and mean latency. Source: results/model_agreement.csv.

Model	n	Strict/Rep./Fail	Usable	ρ	s
gemma4:31b	10	0/10/0	1.00	0.972	2.7
gpt-oss:120b	10	10/0/0	1.00	0.975	6.9
gpt-oss:20b	10	8/2/0	1.00	0.900	9.8
minimax-m2.5	10	8/2/0	1.00	0.994	8.0
minimax-m3	10	6/4/0	1.00	0.991	6.2
nemotron-3-nano:30b	10	10/0/0	1.00	0.978	4.3
nemotron-3-super	10	10/0/0	1.00	0.994	8.5
nemotron-3-ultra	10	9/0/1	0.90	0.987	16.2

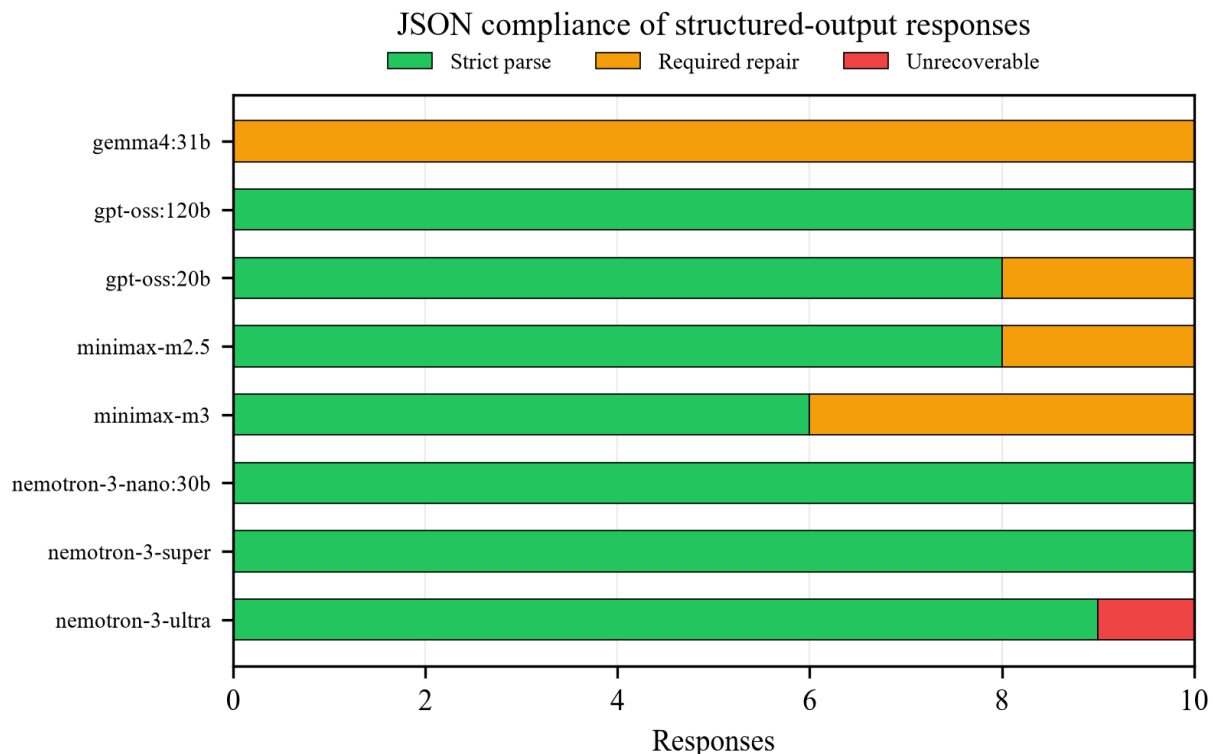


FIGURE 10. Structured-output compliance per model. Repair requirements are strongly model-dependent rather than uniformly distributed.

The aggregated rate hides a very strong bimodal distribution as seen in Fig. 10. Model gemma4:31b gave results that were never strictly

parsable, and always required the repair layer, whereas gpt-oss:120b, nemotron-3-nano:30b and nemotron-3-super were strictly parsable every time.

Similarly, compliance is not a background rate, but a property of a given model, and a deployment that proves the parser to be compliant with one model has proven very little about another. This constitutes one hard evidence for measuring compliance per model when choosing the model, in addition to the quality measurement covered by benchmarks like MTEB [15] and MT-Bench [20].

The agreement between the models on the same pairs was assessed by the standard deviation of the reported similarity across the models for each pair: mean 5.91 percentage points, maximum 16.94.

The overall correspondence between the dense embedding score and the model's reported similarity is 0.9388 in terms of rank. This suggests that the two stages broadly agree on the order of similarity but disagree on the absolute level, which indicates that the dense embedding score can be used as the reported value while the model is used for the explanation. The pattern becomes obvious in Fig. 11: first, the ordering of the tiers is restored in all the models; second, the absolute height of the tiers varies from one model to another.

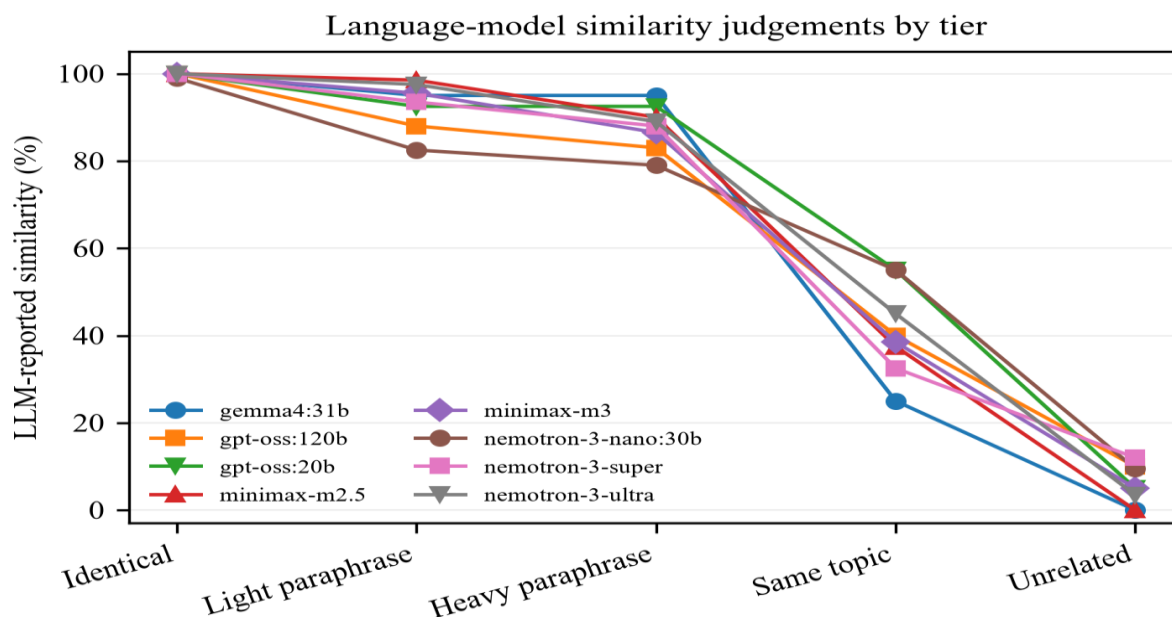


FIGURE 11. Mean similarity reported by each model against ground-truth tier. All models order the tiers correctly; they disagree on absolute values, most visibly at tiers 2 and 1.

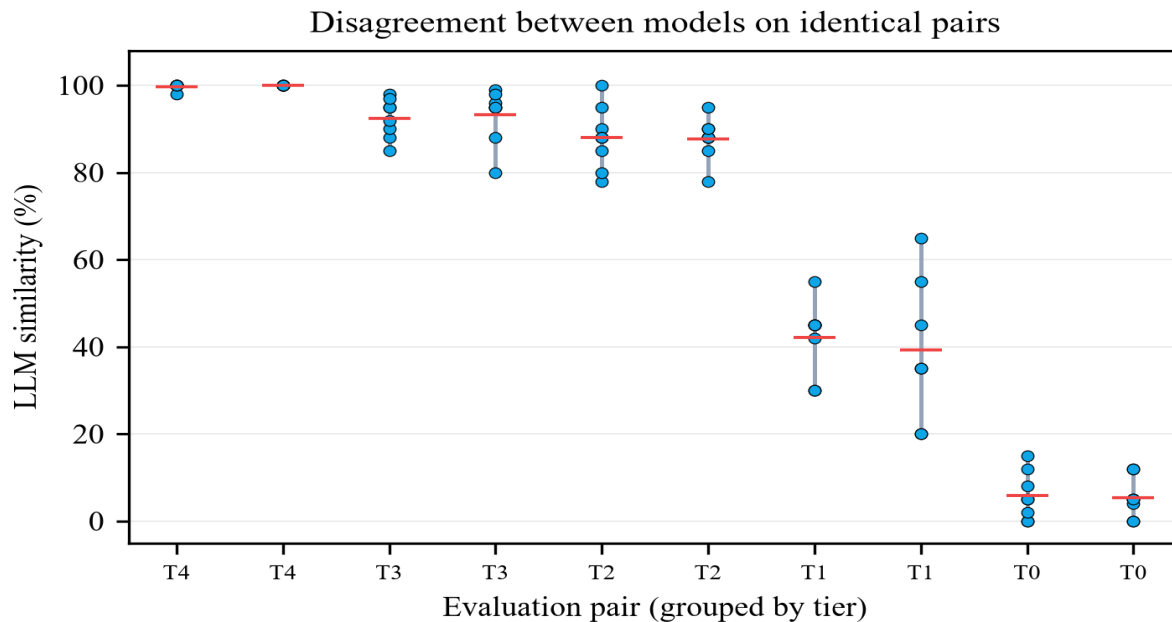


FIGURE 12. Per-pair disagreement between models. Each column is one evaluation pair, labelled by tier; dots are individual model judgements and the red dash is their mean.

The disagreements are seen in pairs in Fig. 12. It is near zero on verbatim copies (where all models score 100) and widens towards the middle of the scale, that is, where judgement is most important and least constrained by the input. A system that reports a single model's number as the headline similarity would then be the most arbitrary on the cases the user would most want to trust, so it is the most substantive reason that the architecture Fig. 1 has the reported score on the deterministic path. That it is being used is proof that it is relevant: a pipeline that expects well-formed JSON would crash every so often on data it didn't control, and those failures would not show up as parse errors, but as missing analysis. For systems that use the

API without any restriction on decoding, post-hoc recovery should be considered a mandatory feature and the rate of its activation should be reported, something not commonly done in the evaluator literature [20, 21].

H. Language-Model Ensembles

In Section IV-G it was established that models disagree in the same inputs. An obvious response to combine them, and because the eight entitled models were each run on the same pairs we can evaluate this without further cost. The individual models are compared to three different combiners – median, mean and trimmed mean (Table 10, Fig. 13).

TABLE 10. Ensembles of 8 language models against individual models. Source: results/metrics.json, v2.e11.

Estimator	rho	AUC	Min gap
ensemble, median	0.9938	1.0000	5.75
ensemble, trimmed mean	0.9877	1.0000	5.75
ensemble, mean	0.9847	1.0000	5.00
best single (minimax-m2.5)	0.9938	1.0000	1.50
worst single (gpt-oss:20b)	0.9000	1.0000	0.00

The median ensemble achieves 0.9938 (vs the best individual model minimax-m2.5, 0.9938, and definitely superior to the worst individual model gpt-oss:20b with 0.9000). This is the true outcome and not that one ensemble folklore suggests:

aggregation is for robustness not for peak accuracy. Its value is that it doesn't need to be known what is the best model; as shown in the entitlement findings of Section IV-G, that knowledge may not last long after the next catalogue change.

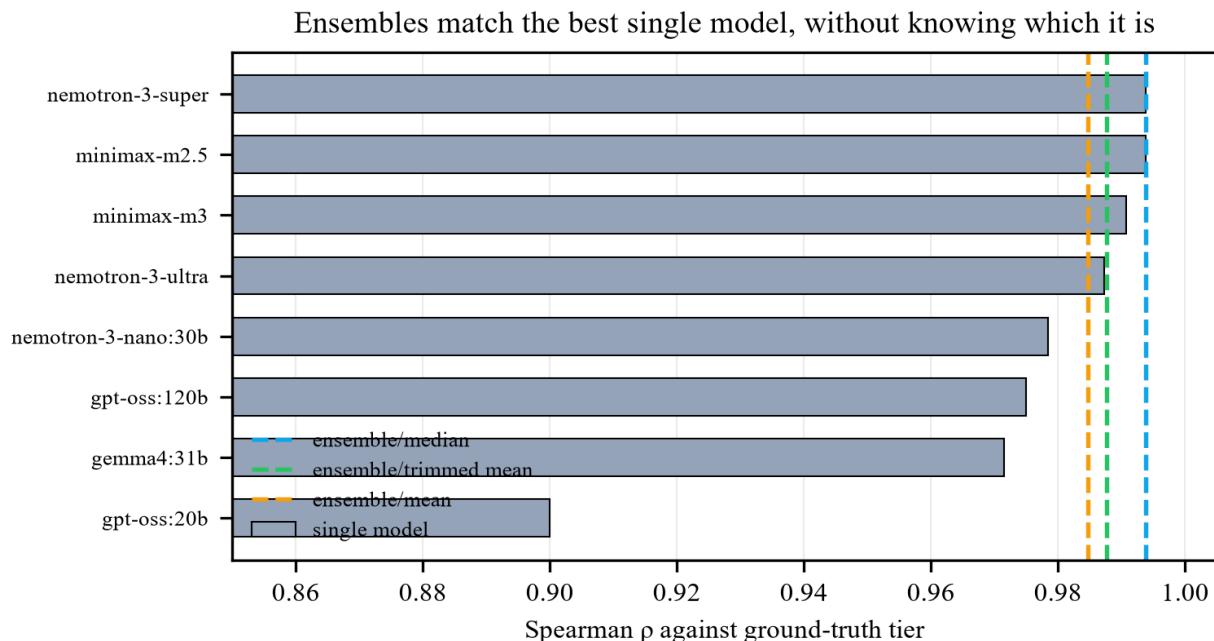


FIGURE 13. Individual models ranked by rank correlation, with the ensemble combiners marked. Ensembles land at the top of the single-model range without identifying it in advance.

The median is more robust than the mean (0.9847), as expected because the one badly-ordered model in the pool. In a deployment where re-benchmarking is not possible when a provider rotates its catalogue, a median over all the models currently entitled is therefore a defensible default (assuming a model chosen correctly is no worse and is roughly 8 times cheaper).

I. Provider Catalogue Versus Actual Entitlement

The generation provider exposes a catalogue endpoint that lists 18 models. When each was probed with a small number of generations requested, it was found that only 8 of these were indeed invocable under the account with a 44% entitlement. The others failed with the distribution of outcomes shown below: 10 not entitled. The partition and detail per model are provided in Fig. 14 and Table 11.

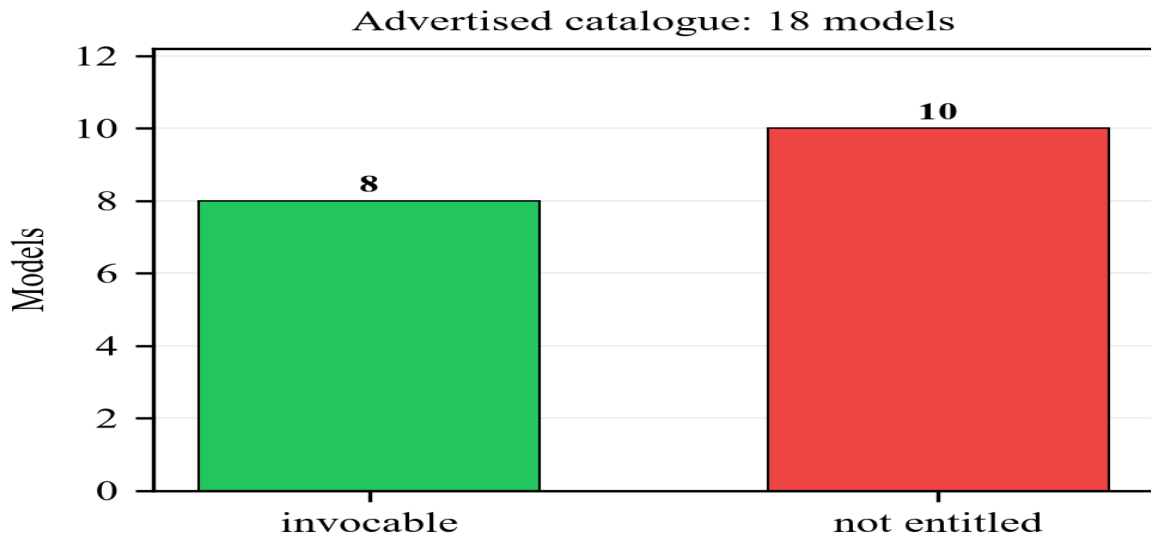


FIGURE 14. Advertised catalogue partitioned by actual invocability under the test account.

TABLE 11. Advertised models and their actual invocability under the test account. Source: results/metrics.json, llm.entitlement_detail.

Model	Invocable	Outcome
deepseek-v4-flash	no	not entitled
deepseek-v4-pro	no	not entitled
gemma4:31b	yes	ok
glm-5.1	no	not entitled
glm-5.2	no	not entitled
gpt-oss:120b	yes	ok
gpt-oss:20b	yes	ok
kimi-k2.5	no	not entitled
kimi-k2.6	no	not entitled
kimi-k2.7-code	no	not entitled
minimax-m2.5	yes	ok
minimax-m2.7	no	not entitled
minimax-m3	yes	ok
mistral-large-3:675b	no	not entitled
nemotron-3-nano:30b	yes	ok
nemotron-3-super	yes	ok
nemotron-3-ultra	yes	ok
qwen3.5:397b	no	not entitled

The outcome is immediate. From the catalogue endpoint, a user interface was created that populated models which were marked to fail and these models were returned as having a similarity of zero as the failure was contained within the analysis

category that had typed defaults. We also noticed a model that was taken out of the system in between releases with a different status code for each request during development. A system that mixes up these

three predicates is periodically and confusingly flawed.

We did two things: converted provider status codes into status messages with a reference to the model and to the solution, and pass on the status codes as warnings containing a clear preservation of the validity of the similarity score to the client. We regard this as the more general lesson of the work in a system composed from third-party inference

services, error provenance is a first-class design concern, not an implementation detail.

J. Cost, Concurrency and Model Selection

The problem of model choice is a trade-off between quality and latency, which is a two-objective problem. The frontier in Table 12 is reported if a model is dominated by a faster and more well ordered model.

TABLE 12. Cost-quality frontier over the entitled models. Latency is the mean per prompt. Source: results/v2_pareto.csv.

Model	Latency (s)	rho	Frontier
gemma4:31b	2.70	0.9715	yes
nemotron-3-nano:30b	4.27	0.9784	yes
minimax-m3	6.20	0.9908	yes
gpt-oss:120b	6.89	0.9750	
minimax-m2.5	7.95	0.9938	yes
nemotron-3-super	8.46	0.9938	
gpt-oss:20b	9.76	0.9000	
nemotron-3-ultra	16.25	0.9873	

There are 4 Pareto-optimal models out of 8. The ramification: The model that comes with this system is dominated – it is slower than four other models, and the worst ordered among all eight. This frontier option, though, yields gemma4:31b at 2.70 s, or a slight loss of time for a little more

ordering. This evidence was being invoked by default when the evidence was originally selected because it was the first one the provider's catalogue listed that the account could invoke. The frontier is shown in Fig. 15.

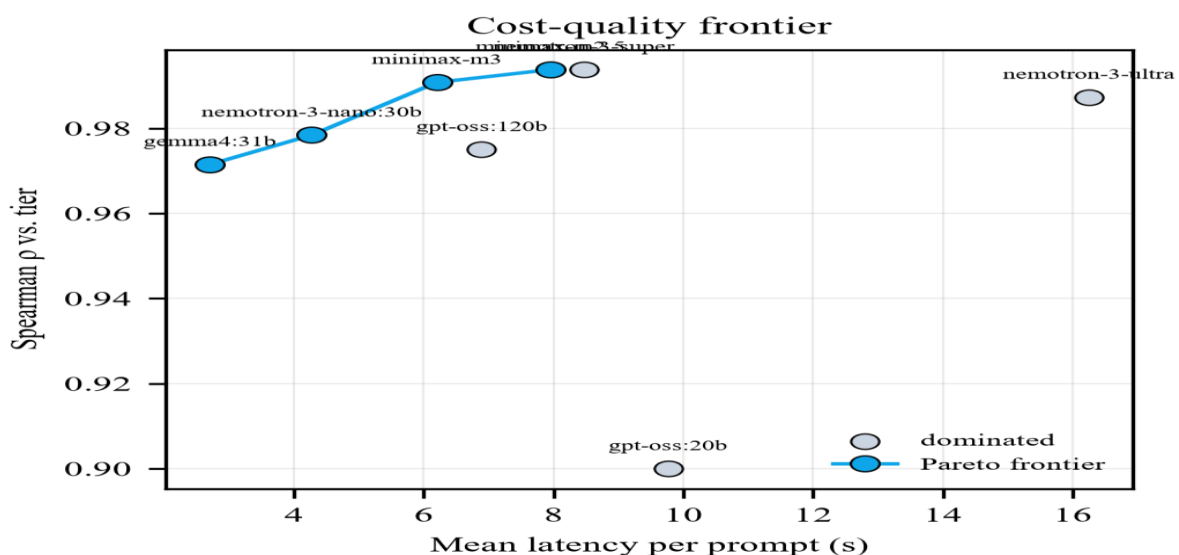


FIGURE 15. Cost-quality frontier. Grey points are dominated; the connected series is the frontier.

According to equation (8), issuing all five independent categories simultaneously should decrease the latency per document by approximately 5 times. It does not. Sequential issuing takes 26.77 s and concurrent issuing 19.57 s, which is 1.37x faster and saves 7.2 s per doc.

TABLE 13. Per-document analysis cost, five independent language-model categories, model nemotron-3-nano:30b. Source: results/v2_concurrency.csv.

Issuing	Mean (s)	SD (s)
sequential	26.77	1.38
concurrent	19.57	3.76

The finding is the shortfall against the 5 fold ideal. The five requests are all independent of each other and so the client sends them out at the same time, so the residual serialisation is provider-side: the client doesn't get five simultaneous generation slots. If the latency budget is specified independently of

their input prompts, practitioners will thus be misled and should measure concurrency actually granted instead of assuming it. The gain is still worthwhile, as this is what is doing to pull a single-document comparison into a serverless execution limit.

serverless
60 s limit

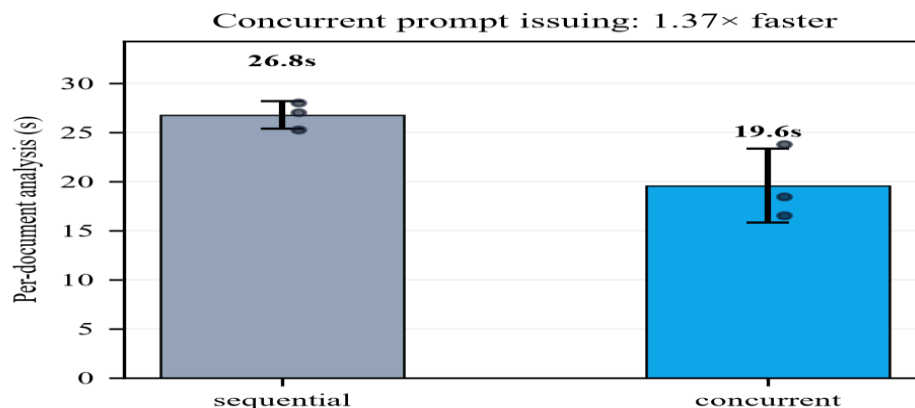


FIGURE 16. Sequential against concurrent issuing of the five independent categories. Dots are individual runs; the dotted line marks a 60-second platform limit.

Here is one methodological note. The earlier iteration of this experiment claimed a result of 3.3x, but it was an artefact because some categories failed when a name-resolution failure occurred during those initial points, but returned the typed default rather than waiting for the provider to return a reply, which prematurely terminated the concurrent condition. With transport-level retries, no failures and the reported speedup was reduced

to the above. We note this because failure mode: a resilience mechanism that is doing a good job of flattering the latency measurement. We note this because it's easy to reproduce and hard to notice.

K. Deployment

The system is publicly deployed with the client as static assets and the analysis tier as a serverless function on the same origin. There are two

platform limitations that were encountered and are worthy of note, the first being due to measurement above and the second due to the configuration preference.

First off is the execution limit. The lowest level of the platform allows only 60 seconds for a function. Sequential issuing: the worst case took 94.2 seconds, making the core feature unusable; Concurrent issuing: a comparison of just a single document was completed within the limit, and the live request measured end-to-end against the deployed instance was returned in 17.1 seconds. Multi-document comparisons are only bound by

the same limit, and are therefore restricted by the deployment, not by the limit.

Second, persistence. The filesystem on the platform is read-only except for a temporary directory that is not preserved between invocations; thus, the JSON store discussed in Section III-M is not a suitable place to store state. As a result, accounts, history and cached vectors will be per-invocation in the public deployment. We believe this would be fine for a demonstrator, but not a viable method of production deployment, and record it as the major hurdle to deployment on this type of platform; it's not a different filesystem strategy that needs to be done.

L. Computational Cost

TABLE 14. Mean per-stage cost. Network-bound stages dominate. Source: results/latency.csv.

Pipeline stage	Mean (ms)
Cohere embedding (per document)	1018.0
Cohere similarity scoring	1023.5
TF-IDF embedding (per document)	3.1
TF-IDF similarity scoring	1.1
Originality (Jaccard + 5-gram)	0.2
LLM semantic prompt (single call)	7810.5

All stages are network bound cost profile. The deterministic lexical measures (9) and (10) take 0.25ms, an order of magnitude less time than a single generation request, with similar exact similarity times for the measures. The number of

provider calls and their concurrency will then be the only lever to play with to optimize the local computation, which would be useless otherwise. The measurements for each stage are shown in Table 14.

TABLE 15. End-to-end latency of complete comparison requests against the running service, model gpt-oss:20b. Source: results/end_to_end.csv.

Docs	LLM calls	n	Mean (s)	s/doc
1	7	3	28.3 ± 5.6	28.3
2	14	3	56.5 ± 9.1	28.3
3	21	3	92.9 ± 1.7	31.0

Table 15 and Fig. 17 are for completed requests, not estimated. The total time for more than 9 requests varied from 21.9 to 94.2 seconds, with a mean of 59.2 seconds. The architecture is linear in the number of comparison documents – the time is

approximately the same for one, two, and three comparison documents, namely 28.3–31.0 seconds per document, each of which incurs 7 sequentially called generations.

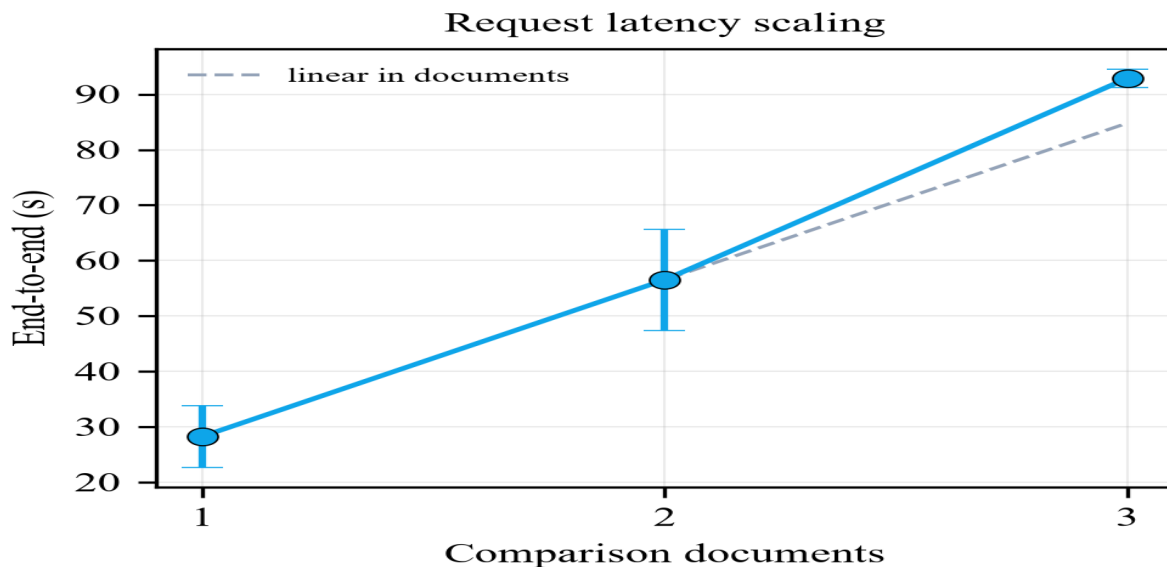


FIGURE 17. End-to-end request latency against the number of comparison documents. Error bars are one standard deviation; the dashed line is a linear reference through the origin.

This has an impact on deployment, worthy of note. The analysis tier has a long running host requirement because, although it has modest compute requirements, it is being called when there is a measured worst case of 94.2 seconds, which is longer than the execution time limits of common serverless function platforms (typically 10 seconds on the entry tiers, and 60 seconds on paid tiers). The entry-tier limit is being exceeded as it takes only 28.3 seconds to compare two documents. Each prompt per document is independent and thus, when issued all at once, it should reduce the time for each per-document comparison down to roughly the slowest per-document prompt, making each document comparison reachable within a 60-second document prompt budget; this optimization is currently not yet applied.

M. Discussion

There are three generalisations that can be made from this system. First, it is important that a hybrid architecture not view fallback as an option as an interchangeable component. The lexical fallback studied here is valid on all interfaces but removes ordinal validity, and the type checking would not complain about that. When it is not possible to

make the calibration of a fallback, it should be declared to the consumer; a fallback method that is calibrated onto the primary path's scale is a better option.

Secondly, there is a cautionary case to be made about the composite readability score. The implementation calculates quality based on Flesch reading ease penalised by the grade above 8th grade, which assigns a quality of zero to any academic text that is dense, and rates it as the lowest available band. All of the reports written in this evaluation were very easy to find the lowest possible mark on each individual report in the corpus, which consisted of competent expository writing on technical topics. The formula reflects reading ease, since its components were designed to [25]–[30] and a rephrased version of the formula as quality would suggest is misleading for the academic audience that the system targets. If we don't report this, we are indirectly fixing this instead of making it an explicit design change for what the system should say it measures.

Third, operational reliability is a research issue and not just an engineering nuisance with respect to hosted inference. When a model identifier is given in a reproducibility claim, it must be served and the

reader must have access to it, which we saw is not the case in a single development cycle. Reporting LLM-based results should include the access tier used and date of access and, ideally, report behaviour under provider failure.

N. Guidance for Practitioners

The above results are summarized here as five recommendations which would be used for a similar system.

Maintain a single correct metric of the report. Find the value of the headline number using a constant function of the input. Use the language model only for things that can't be expressed by a scalar. This allows the number to be reproduced and the system to not be completely useless if there is a failure in generation.

Don't let a back-up go unspoken. A degraded path which fulfils the same interface will be accepted as the primary path. Either calibrate it onto the primary scale, label its output at the point of presentation, fail closed. The middle option is our measurement of the minimum acceptable standard and uninformative over half scales.

Count the percentage of structured output that is correct for each model. However, compliance was not uniform and a parser that is validated to one model will not give you any information about another. When changing model, re-measure the compliance rate and incorporate that in model-selection criteria.

Check for eligibility – not catalogue membership. Test each advertised model with a small request prior to offering it to users and cache the result. A catalogue endpoint is not a permission item associated with your account, but rather the provider's inventory.

Translate boundary provider errors. A raw status code cascades inwards to default value and plausible-looking result. An outage becomes a diagnosis once it is translated by the client boundary, to message naming the resource and the remedy.

O. Limitations and Threats to Validity

The size and architecture of the corpora. The Corpus consists of 5 topics with 30 pairs written by

one writer. Well-separated tiers are better suited to describe perfect separation on the derivation-detection task, and the lack of inter-annotator agreement statistics is a major drawback. Replication would be on an established benchmark like the PAN corpora [6] which would lend more external validity.

This information will be added to the length of the document and the multi-chunk sub-corpus. In the primary corpus, the average length of each document is 130.4 words, for which (1) and the pooling in (2) are not exercised; hence they reduce to one value, Section IV-D. The multi-chunk sub-corpus is only partially able to address this because it only has two to three chunks per document, and it is not separated by length, but deals with shared boilerplate also. Solid: max-pooling collapses if there is boilerplate shared across the length of the study, clean length scaling study is still owed.

Sample size for second round comparisons. In the representation study, the 30 pairs are used in the same way as in the aggregation study, and in the same way as in the repair study. Differences less than a point in the minimum adjacent gap are within the noise of that sample; and we have specifically stated at each point what we're comparing and what we're not. These large effects (monotonicity restored, IDF harmful, max-pooling collapsing) are way outside.

Language and register. All documents are written expository prose in English. The embedding model of the program is clearly an English-only model, and neither multilingual behaviour nor the cross-lingual reuse that plagiarism detection is frequently required to deal with is tested.

Provider dependence. The results depend on the specific versions of the models that are hosted at a particular time and the results shown as entitlement findings are properties of one account tier. Qualitative conclusions to fall-back validity and failure provenance should not be expected to change, while the absolute values should.

No human reference. The ground truth is not judges generated, but is a set of orders that are derived from the constructed system; it is established that the system can recover the ordering that it was built to recover, and not that it matches

human similarity judgement. The most crucial future work is that comparison.

P. Concluding Remarks

We proposed a system for comparing documents based on both quantitative similarity score based on dense embeddings and qualitative analysis based on a language model and tested it on 30 handwritten pairs of documents with different labels. The dense path performs with a high degree of accuracy— $\rho = 0.975$ —on this corpus and is able to separate derived sentences that are verbatim reuses from those that are not, due to its high precision rating of the verbatim reuse evidence provided by the deterministic five-gram measure.

What we are interested in that is more transferable, however, has to do with failure, not accuracy; method used find. The adjacent-gap analysis reported it and the tier means are not even weakly ordered, and, unweighted term frequency assigned it, so we can now confirm that the repair was effective. Negative result local inverse document frequency, decoupling of embedding size from accuracy, and collapse of max-pooling under shared boilerplate – the three findings that these correlations aggregate and hide altogether, as all configurations score the same 0.9745 on this statistic.

In addition to these, the operational conclusions are each a plausible-looking number unless they are explicitly dealt with, namely a provider catalogue with models being advertised, structured output where the repair requirement depends on the model bimodally, and a concurrency gain that is at least a third of what task independence suggests. The challenge of designing in such a way that such

a condition emerges as a diagnosis, not as data, is what we believe is the main task of systems where inference services are third parties.

Two areas that were identified in the first round of this study were completed and are reported above: (a) concurrent prompt issuing, which sufficiently lowered the per-document latency to allow the comparison to be performed in a serverless execution time, and (b) isotonic calibration of the fallback to the dense scale, which achieved a mean absolute error of 9.44 points. Both are reported with the caveats that accompany their measurements.

The remainder is mainly about scale and external validity. Replication against the PAN corpora and against elicited human similarity judgements would confirm that the system is able to recover human notions of similarity rather than just the ordering it was built to recover. A length-scaling study conducted on a clean corpus would isolate document length from common boilerplate, as it does in the present sub-corpus. The obvious candidate is the boilerplate problem, where one might want to down-weight passages that are similar to many passages in the training set, but the Inverse Document Frequency (IDF) result of Section IV-E suggests that this does not necessarily help.

Q. Reproducibility

Every figure and table in this section is generated from artifacts produced by three commands. Table 16 maps each result to its source file. The document itself is assembled programmatically from results/metrics.json, so no reported number is transcribed by hand.

TABLE 16. Mapping from reported results to source artifacts.

Result	Artifact
Table 1	evaluation/corpus.py
Tables 2, 7; Figs. 2, 3, 8	similarity_scores.csv
Tables 3, 4; Fig. 4	v2_embedding_models.csv; v2_input_types.csv
Table 5; Fig. 5	v2_long_documents.csv
Table 6; Figs. 6, 7	v2_lexical_variants.csv
Table 8; Fig. 9	originality_scores.csv
Table 9; Figs. 10, 11, 12	model_agreement.csv
Table 10; Fig. 13	metrics.json (v2.e11)

Table 11; Fig. 14	metrics.json (llm.entitlement_detail)
Table 12; Fig. 15	v2_pareto.csv
Table 13; Fig. 16	v2_concurrency.csv
Table 14	latency.csv
Table 15; Fig. 17	end_to_end.csv
Fig. 1	evaluation/run_evaluation.py
Experiments E1-E6	evaluation/run_evaluation.py
Experiments E7-E14	evaluation/experiments_v2.py
Figures E7-E14	evaluation/figures_v2.py
This document	paper/build_paper.py

V. REFERENCES

- [1] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira, "Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations," in Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2021, pp. 2356-2362, doi: 10.1145/3404835.3463238.
- [2] N. Thakur, N. Reimers, A. Rueckle, A. Srivastava, and I. Gurevych, "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in Advances in Neural Information Processing Systems, vol. 34, 2021.
- [3] T. Formal, B. Piwowarski, and S. Clinchant, "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking," in Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2021, pp. 2288-2292, doi: 10.1145/3404835.3463098.
- [4] R. Tripodi, R. Blloshmi, and S. Levis Sullam, "Evaluating Multilingual Sentence Representation Models in a Real Case Scenario," in Proc. 13th Language Resources and Evaluation Conf. (LREC), Marseille, France, 2022, pp. 2928-2939.
- [5] J. P. Wahle, T. Ruas, F. Kirstein, and B. Gipp, "How Large Language Models are Transforming Machine-Paraphrase Plagiarism," in Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP), Abu Dhabi, United Arab Emirates, 2022, pp. 952-963, doi: 10.18653/v1/2022.emnlp-main.62.
- [6] J. Lee, T. Agrawal, A. Uchendu, T. Le, J. Chen, and D. Lee, "PlagBench: Exploring the Duality of Large Language Models in Plagiarism Generation and Detection," in Proc. 2025 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Albuquerque, NM, USA, 2025, pp. 7519-7534, doi: 10.18653/v1/2025.naacl-long.384.
- [7] A. Kusupati et al., "Matryoshka Representation Learning," in Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 30233-30249.
- [8] H. Su et al., "One Embedder, Any Task: Instruction-Finetuned Text Embeddings," in Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, 2023, pp. 1102-1121, doi: 10.18653/v1/2023.findings-acl.71.
- [9] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving Text Embeddings with Large Language Models," in Proc. 62nd Annu. Meeting of the Association for Computational Linguistics (ACL), Bangkok, Thailand, 2024, pp. 11897-11916, doi: 10.18653/v1/2024.acl-long.642.

- [10] H. W. Chung et al., "Scaling Instruction-Finetuned Language Models," *J. Mach. Learn. Res.*, vol. 25, no. 70, pp. 1-53, 2024.
- [11] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei, "Text Embeddings by Weakly-Supervised Contrastive Pre-training," *arXiv:2212.03533*, 2022, doi: 10.48550/arXiv.2212.03533.
- [12] K. Wang, N. Reimers, and I. Gurevych, "TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, 2021, pp. 671-688, doi: 10.18653/v1/2021.findings-emnlp.59.
- [13] J. Ni et al., "Large Dual Encoders Are Generalizable Retrievers," in *Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi, United Arab Emirates, 2022, pp. 9844-9855, doi: 10.18653/v1/2022.emnlp-main.669.
- [14] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic, 2021, pp. 6894-6910, doi: 10.18653/v1/2021.emnlp-main.552.
- [15] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive Text Embedding Benchmark," in *Proc. 17th Conf. European Chapter of the Association for Computational Linguistics (EACL)*, Dubrovnik, Croatia, 2023, pp. 2014-2037, doi: 10.18653/v1/2023.eacl-main.148.
- [16] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535-547, 2021, doi: 10.1109/TBDATA.2019.2921572.
- [17] A. Rege et al., "AdANNS: A Framework for Adaptive Semantic Search," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 76311-76335.
- [18] G. Izacard et al., "Atlas: Few-shot Learning with Retrieval Augmented Language Models," *J. Mach. Learn. Res.*, vol. 24, no. 251, pp. 1-43, 2023.
- [19] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models Are Zero-Shot Learners," in *Proc. 10th Int. Conf. Learning Representations (ICLR)*, 2022.
- [20] L. Zheng et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 46595-46623.
- [21] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation Using GPT-4 with Better Human Alignment," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2023, pp. 2511-2522, doi: 10.18653/v1/2023.emnlp-main.153.
- [22] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824-24837.
- [23] W. Yuan, G. Neubig, and P. Liu, "BARTScore: Evaluating Generated Text as Text Generation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 27263-27277.
- [24] A. R. Fabbri et al., "SummEval: Re-evaluating Summarization Evaluation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 391-409, 2021, doi: 10.1162/tacl_a_00373.
- [25] B. W. Lee, Y. S. Jang, and J. Lee, "Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features," in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 10669-10686, doi: 10.18653/v1/2021.emnlp-main.834.
- [26] S. A. Crossley, A. Heintz, J. S. Choi, J. Batchelor, M. Karimi, and A. Malatinszky, "A large-scaled corpus for assessing text readability," *Behav. Res. Methods*, vol. 55, no. 2, pp. 491-507, 2023, doi: 10.3758/s13428-022-01802-x.

- [27] L. Gaeta, E. Garcia, and V. Gonzalez, "Readability and Suitability of Spanish-Language Hearing Aid User Guides," *Am. J. Audiology*, vol. 30, no. 2, pp. 452-457, 2021.
- [28] E. Nash, M. Bickerstaff, A. J. Chetwynd, D. B. Hawcutt, and L. Oni, "The readability of parent information leaflets in paediatric studies," *Pediatr. Res.*, vol. 94, no. 3, pp. 1166-1171, 2023, doi: 10.1038/s41390-023-02608-z.
- [29] S.-S. Ucar, I. Aldabe, N. Aranberri, and A. Arruarte, "Exploring Automatic Readability Assessment for Science Documents within a Multilingual Educational Context," *Int. J. Artif. Intell. Educ.*, vol. 34, pp. 1417-1459, 2024, doi: 10.1007/s40593-024-00393-2.
- [30] B. W. Lee and J. H.-J. Lee, "Traditional Readability Formulas Compared for English," *arXiv:2301.02975*, 2023, doi: 10.48550/arXiv.2301.02975.
- [31] Y. Li, Z. Wang, L. Cui, W. Bi, S. Shi, and Y. Zhang, "Spotting AI's Touch: Identifying LLM-Paraphrased Spans in Text," in *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, 2024, pp. 7088-7107, doi: 10.18653/v1/2024.findings-acl.423.
- [32] A. Lovakov and E. R. Agadullina, "Empirically derived guidelines for effect size interpretation in social psychology," *Eur. J. Soc. Psychol.*, vol. 51, no. 3, pp. 485-504, 2021, doi: 10.1002/ejsp.2752.
- [33] S. K. Corbacioglu and G. Aksel, "Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value," *Turk. J. Emerg. Med.*, vol. 23, no. 4, pp. 195-198, 2023, doi: 10.4103/tjem.tjem_182_23.
- [34] M. Hassanzad and K. Hajian-Tilaki, "Methods of determining optimal cut-point of diagnostic biomarkers with application of clinical data in ROC analysis: an update review," *BMC Med. Res. Methodol.*, vol. 24, Art. no. 84, 2024, doi: 10.1186/s12874-024-02198-2.
- [35] A. Michail, S. Clematide, and J. Opitz, "PARAPHRASUS: A Comprehensive Benchmark for Evaluating Paraphrase Detection Models," in *Proc. 31st Int. Conf. Computational Linguistics (COLING)*, Abu Dhabi, United Arab Emirates, 2025, pp. 8749-8762.
- [36] M. Andrecut, "Additive Feature Hashing," *arXiv:2102.03943*, 2021, doi: 10.48550/arXiv.2102.03943.
- [37] T. Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: a survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211-3260, 2023, doi: 10.1007/s10994-023-06336-7.
- [38] A. Henzi, A. Moesching, and L. Duembgen, "Accelerating the Pool-Adjacent-Violators Algorithm for Isotonic Distributional Regression," *Methodol. Comput. Appl. Probab.*, vol. 24, no. 4, pp. 2633-2645, 2022, doi: 10.1007/s11009-022-09937-2.
- [39] G. H. Chen, S. Chen, Z. Liu, F. Jiang, and B. Wang, "Humans or LLMs as the Judge? A Study on Judgement Bias," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Miami, FL, USA, 2024, pp. 8301-8327, doi: 10.18653/v1/2024.emnlp-main.474.
- [40] Q. Pan et al., "Human-Centered Design Recommendations for LLM-as-a-Judge," in *Proc. 1st Workshop on Human-Centered Large Language Modeling (HuCLLM)*, St. Julian's, Malta, 2024, pp. 16-29, doi: 10.18653/v1/2024.hucllm-1.2.
- [41] Z. R. Tam et al., "Let Me Speak Freely? A Study on the Impact of Format Restrictions on Large Language Model Performance," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing: Industry Track*, Miami, FL, USA, 2024, pp. 1218-1236, doi: 10.18653/v1/2024.emnlp-industry.91.

- [42] S. Geng, M. Josifoski, M. Peyrard, and R. West, "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning," in Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP), Singapore, 2023, pp. 10932-10952, doi: 10.18653/v1/2023.emnlp-main.674.
- [43] Y. Lu et al., "Learning to Generate Structured Output with Schema Reinforcement Learning," in Proc. 63rd Annu. Meeting of the Association for Computational Linguistics (ACL), Vienna, Austria, 2025, pp. 4905-4918, doi: 10.18653/v1/2025.acl-long.243.
- [44] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," Eng. Appl. Artif. Intell., vol. 115, Art. no. 105151, 2022, doi: 10.1016/j.engappai.2022.105151

