International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

A Hybrid Semantic and Lexical Framework for Enterprise Document Similarity Detection

Srikanth Dokku[1], Dr. Ranjith Gopalan[2], ManmadhaRao Baipalli[3]
[1]*Gen AI Tech Lead, Capgemini*
[2]*Principal Consultant, Cognizant Technology Solutions*
[3]*Gen AI Solutions Architect, Capgemini.*

***Abstract*-- Enterprise organizations manage large and continuously expanding collections of contracts, resumes, invoices, policies, technical documents, and compliance records that evolve through revisions, approvals, and multi-team collaboration. Traditional similarity systems based solely on lexical matching often fail to identify semantically related documents when wording changes, while embedding-only approaches may miss structural or formatting-based similarities that are operationally important in legal and compliance contexts. This paper presents a hybrid framework that integrates semantic embeddings (Sentence-BERT, cosine similarity), lexical techniques (trigram analysis, token overlap scoring, Jaccard similarity), TF-IDF weighted retrieval, BM25 ranking, and structural document analysis for enterprise-grade document similarity detection. The framework is designed to improve retrieval quality, reduce duplicate content propagation, and provide explainable decision support for audit-sensitive workflows. Empirical evaluation on a curated benchmark of 1,000 enterprise documents and 4,800 labeled pairs demonstrates an F1-score of 0.86, surpassing single-method baselines by margins of 7–16 percentage points. The architecture is further validated through integration with a Retrieval-Augmented Generation (RAG) pipeline, confirming measurable improvements in retrieval precision and reduced context redundancy.**

***Keywords*-- Document Similarity, Explainable AI, Hybrid Retrieval, Semantic Embeddings, Lexical Analysis, Enterprise Knowledge Systems, BM25, Jaccard Similarity, Near-Duplicate Detection, RAG**

I. Introduction

Document similarity detection is a foundational requirement in modern enterprise information systems. Organizations across industries — financial services, healthcare, legal, manufacturing, and government — manage repositories that can contain millions of documents accumulated over decades of operations. These repositories are not static: documents are drafted, revised, reviewed, approved, archived, and periodically superseded by updated versions. In high-volume environments, near-duplicates create operational friction, increase review effort, and introduce risk when outdated or partially revised versions are inadvertently reused or distributed.

The challenge of document similarity in enterprise settings is qualitatively different from academic information retrieval benchmarks. Enterprise documents share vocabulary within domains (legal, financial, HR) yet differ critically at the clause level. A contract that changes a single payment deadline or a policy that updates a liability threshold may appear nearly identical to a lexical matcher, yet the business impact of the change is significant. Conversely, two policy documents drafted independently in different departments may use entirely different phrasing to communicate the same organizational intent — a scenario where pure keyword overlap fails and semantic understanding is necessary.

The volume of unstructured enterprise content has grown substantially in recent years. Industry reports suggest that enterprises with over 10,000 employees generate upwards of 100,000 new documents per year across contracts, HR records, compliance filings, and operational manuals [IDC, 2023]. Without robust deduplication and similarity detection, knowledge management platforms accumulate redundant content that degrades search quality, consumes storage, and increases the cognitive load on document reviewers.

Traditional keyword search approaches, including TF-IDF and BM25, excel at matching exact or near-exact phrasing but are brittle to synonymy and paraphrasing. Purely semantic embedding systems — built on transformer architectures such as BERT and Sentence-BERT — provide strong robustness to lexical variation by operating in a high-dimensional meaning space, but may assign high similarity scores to documents that use equivalent language while differing in legally significant structural details such as clause ordering, table contents, or formatting conventions. An enterprise-ready solution therefore requires balanced sensitivity: the ability to identify paraphrastic equivalence while preserving precision on clause-level differences that influence business outcomes.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Regulatory and compliance pressures further motivate this research. Frameworks such as GDPR (General Data Protection Regulation), SOX (Sarbanes-Oxley Act), and HIPAA require organizations to maintain version-controlled, auditable document trails. Automated similarity detection supports these requirements by flagging near-duplicate policy updates, identifying version lineage in contract repositories, and surfacing unauthorized copies of sensitive documents. In such contexts, explainability — the ability to articulate why two documents were deemed similar — is as important as accuracy.

This paper makes the following contributions:

1. A five-layer hybrid architecture combining semantic, lexical, set-based, token frequency, and structural similarity signals into a unified weighted scoring framework.
2. A weighted ensemble scoring algorithm (Algorithm 1) with tunable weights per document category, enabling domain-specific calibration.
3. An explainability layer that surfaces per-signal contributions, matched evidence spans, and confidence estimates alongside every similarity decision.
4. Empirical validation on a 4,800-pair enterprise benchmark with ablation analysis confirming the incremental value of each layer.
5. A design blueprint for integrating hybrid similarity into RAG pipelines as both a pre-retrieval deduplication gate and a post-retrieval reranking signal.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 formalizes the problem statement. Section 4 describes the proposed hybrid architecture. Section 5 details the similarity algorithms. Section 6 presents enterprise use cases. Section 7 describes the explainability framework. Section 8 reports experimental results. Section 9 covers RAG integration. Section 10 discusses challenges and limitations. Section 11 outlines future work, and Section 12 concludes.

II. Background And Related Work

Prior research on document similarity and near-duplicate detection spans several decades and encompasses lexical, statistical, and neural approaches. This section surveys the most relevant lines of work and positions the proposed framework within the existing landscape.

2.1 Lexical and Statistical Methods

The earliest and most widely deployed approaches to document similarity rely on lexical overlap. **Term Frequency–Inverse Document Frequency (TF-IDF)**[Salton & McGill, 1983] represents each document as a sparse vector of weighted term frequencies, where weights discount terms that appear frequently across the corpus (low discriminative value) and amplify rare but informative terms. Cosine similarity between TF-IDF vectors provides an efficient pairwise comparison metric that scales well to large corpora and has been a production standard in enterprise search platforms for over three decades.

BM25(Best Match 25) [Robertson & Zaragoza, 2009] extends TF-IDF by incorporating document length normalization and a saturation function on term frequency, making it more robust to document length variation. BM25 has become the default ranking function in systems such as Elasticsearch and Apache Solr and remains a strong baseline in competitive information retrieval evaluations [Thakur et al., 2021].

N-gram and trigram overlap methods measure similarity at the character or word sequence level. Character-level n-grams are particularly robust to morphological variation, spelling errors, and OCR artifacts that are common in scanned enterprise documents [Manning et al., 2008]. Jaccard similarity over n-gram sets provides a normalized overlap coefficient that is insensitive to document length and straightforward to interpret.

Edit distance(Levenshtein distance) [Levenshtein, 1966] quantifies the minimum number of character-level insertions, deletions, and substitutions required to transform one string into another. While computationally expensive for long documents (O(mn) time), edit distance remains useful for short text fields such as document titles, headers, and clause identifiers.

MinHash and Locality-Sensitive Hashing (LSH)[Broder, 1997; Indyk & Motwani, 1998] offer sub-linear approximate nearest-neighbor search for Jaccard similarity by compressing set membership into compact signature matrices. MinHash-based systems such as Google's web-scale near-duplicate detection pipeline [Charikar, 2002] demonstrated that fingerprint-based deduplication can scale to billions of documents. SimHash [Charikar, 2002] generalizes this principle to cosine similarity and is widely used in web crawl deduplication.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

2.2 Word and Sentence Embeddings

The introduction of **Word2Vec**[Mikolov et al., 2013] and **GloVe**[Pennington et al., 2014] marked a shift toward distributed representations that encode semantic relatedness. Document-level embeddings constructed by averaging word vectors provided a simple but effective baseline for semantic similarity that outperformed TF-IDF cosine on paraphrase detection tasks [Wieting et al., 2015].

FastText[Bojanowski et al., 2017] extended Word2Vec by incorporating subword information, improving representations for morphologically rich languages and out-of-vocabulary terms, which is particularly relevant for enterprise corpora containing technical jargon, product codes, and domain-specific abbreviations.

The advent of transformer-based language models beginning with **BERT**[Devlin et al., 2019] enabled contextualized representations in which the embedding of a word depends on its full sentential context. This substantially improved semantic similarity performance on benchmarks such as STS (Semantic Textual Similarity) [Cer et al., 2017] and SNLI [Bowman et al., 2015]. However, BERT's [CLS] token representation is not directly optimized for semantic similarity, requiring fine-tuning on sentence-pair tasks for competitive performance.

Sentence-BERT (SBERT)[Reimers & Gurevych, 2019] addresses this limitation by fine-tuning siamese and triplet BERT networks on sentence-pair datasets to produce dense sentence embeddings optimized for cosine similarity comparison. SBERT representations are highly effective for semantic textual similarity, paraphrase detection, and information retrieval, and have become the de facto embedding backbone for enterprise semantic search pipelines.

More recent models including **E5**[Wang et al., 2022], **BGE**[Xiao et al., 2023], and **OpenAI text-embedding-3**[OpenAI, 2024] have extended embedding performance further through instruction-tuned training on large-scale retrieval datasets. These models demonstrate strong zero-shot generalization across diverse document types, making them suitable candidates for enterprise similarity systems without domain-specific fine-tuning.

2.3 Hybrid and Re-ranking Approaches

Recognition that neither lexical nor semantic methods alone are sufficient has motivated a body of work on hybrid retrieval. **Dense Passage Retrieval (DPR)** [Karpukhin et al., 2020] demonstrated that learned dense retrievers outperform BM25 on open-domain question answering, but subsequent work showed that combining DPR with BM25 via score interpolation produces further gains [Luan et al., 2021].

The **SPLADE** family [Formal et al., 2021] learns sparse representations that capture term importance with semantic awareness, bridging the gap between lexical and dense methods.

In enterprise search, re-ranking approaches apply a cross-encoder model to the top-K candidates retrieved by a first-stage retriever, exploiting the deeper contextual interaction between query and document [Nogueira & Cho, 2019]. While effective, cross-encoders are computationally expensive and not suitable as primary similarity scorers for pairwise comparison at scale.

Recent enterprise RAG deployments [Gao et al., 2023; Edge et al., 2024] indicate that neither purely dense retrieval nor lexical retrieval is consistently sufficient in isolation. Hybrid retrieval combining BM25 and dense embeddings with reciprocal rank fusion (RRF) has emerged as a practical default in production RAG systems.

2.4 Document Structure and Metadata Similarity

Document structure carries semantic signals that are orthogonal to textual content. Research on XML and HTML structure comparison [Cobena et al., 2002; Wang et al., 2003] established tree edit distance as a formal measure for structural similarity. In enterprise document analysis, structural features such as section ordering, heading hierarchies, table presence, and template adherence provide important signals for version detection and duplicate classification that text-based methods miss. The proposed framework incorporates structure analysis as a distinct scoring layer, drawing on these observations.

III. Problem Statement: Why Hybrid Similarity Is Needed

3.1 Formal Problem Definition

Let $D = \{d_1, d_2, \ldots, d_n\}$ denote a corpus of enterprise documents. Each document d_i is characterized by its textual content T_i, structural representation Str_i (section hierarchy, headings, tables), and metadata M_i (author, date, document type, version).

The **document similarity problem** is to define a function:

such that S is high when two documents are semantically and/or lexically similar, and low otherwise, with the additional requirements that:

1. *Calibration:* scores are interpretable across document types and length ranges.
2. *Explainability:* the contributing factors to S are decomposable and presentable to human reviewers.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

3. *Robustness:* scores are stable under OCR noise, reformatting, and minor lexical variation that does not alter document intent.
4. *Efficiency:* pairwise scoring latency supports near-real-time or batch production workloads.

3.2 Taxonomy of Enterprise Document Similarity

Enterprise similarity cases fall into a four-level taxonomy based on the nature and degree of overlap:

Level	Type	Description	Primary Detection Signal
L1	Exact Duplicate	Byte-for-byte or near-byte identical copies	Hash / fingerprint
L2	Near-Duplicate	Minor edits: date changes, name substitutions, reformatting	Trigram / Jaccard / token overlap
L3	Paraphrastic Duplicate	Same intent, different wording (template rewrites, translations)	Semantic embedding cosine
L4	Functionally Similar	Same domain and purpose, different clauses or terms	Hybrid (semantic + structural)

No single method reliably handles all four levels. Exact fingerprinting handles L1 trivially but fails at L2–L4. Lexical overlap handles L2 well but degrades at L3–L4. Semantic embeddings handle L3 but may over-score L4 cases where structural or clause-level differences are critical. A hybrid system is required to provide reliable performance across the full taxonomy.

3.3 Failure Modes of Single-Method Systems

Scenario	Embedding-Only Risk	Lexical-Only Risk	Hybrid Benefit
Clause reorder in contracts	May overestimate similarity despite compliance-critical order changes	May over-penalize reorder with same meaning	Combines semantic equivalence with structural checks
OCR noise in scanned documents	Unstable embeddings from corrupted text	Sharp token mismatch from OCR errors	Normalization plus mixed signals improve robustness
Paraphrased policy duplicates	Good semantic capture	Low overlap despite same policy intent	Semantic score preserves match while lexical verifies key terms
Template-based documents	High scores across all instances regardless of filled values	Catches filled-in value differences	Structure similarity identifies template family; lexical distinguishes instances
Version tracking (v1 vs v2)	High similarity may mask critical changes	Change-sensitive but noisy on minor reformatting	Hybrid delta score highlights changed sections with evidence

IV. Proposed Hybrid Architecture

The proposed system comprises five major processing layers that operate as a coordinated pipeline from document ingestion to explainable ranking. Each layer produces intermediate signals that are retained for transparency, enabling reviewers to interpret not only the final score but also the evidence supporting that score.

The architecture is designed with modularity in mind: individual layers can be enabled, disabled, or weight-adjusted based on document type and operational requirements.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

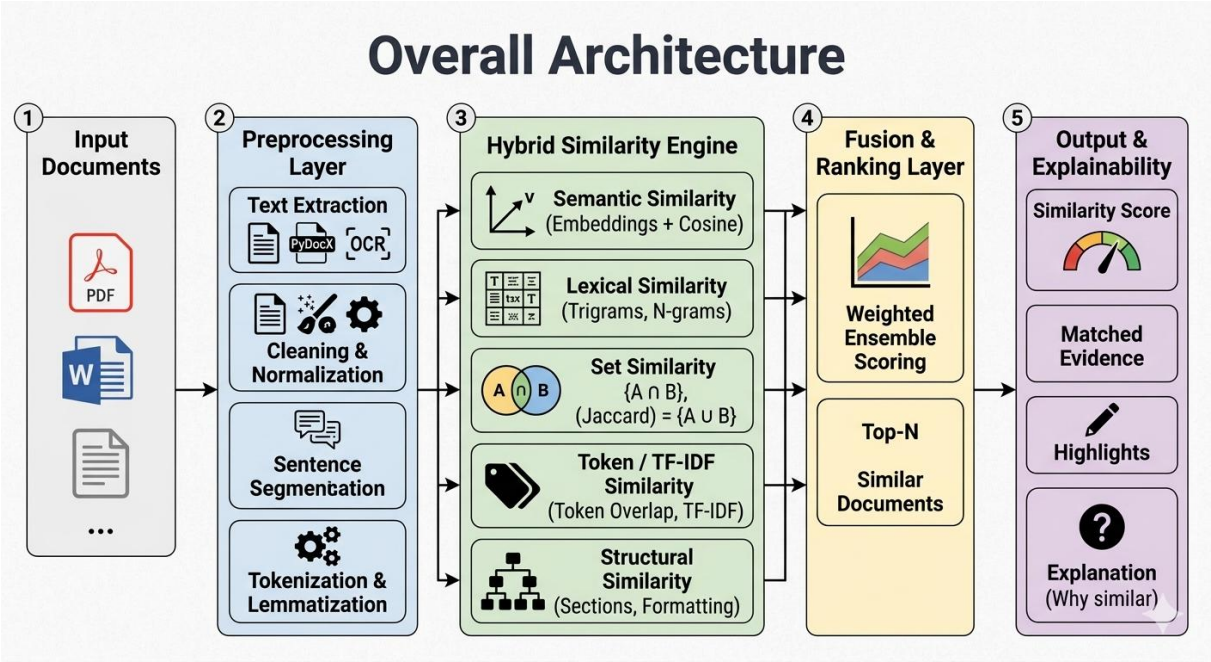


Figure 1. Overall five-stage architecture, from heterogeneous document ingestion and preprocessing through hybrid scoring, ranking, and explainable output.

4.1 Layer 1: Document Extraction and Preprocessing

The extraction layer converts heterogeneous document formats (PDF, DOCX, TXT, HTML, XLSX) into normalized Unicode text streams with preserved structural metadata. Key operations include:

- *Format-aware parsing:* DOCX files are parsed using python-docx to preserve paragraph styles, heading levels, table structures, and list formatting. PDF files are processed through a two-pass pipeline: PyMuPDF for native PDF text extraction, followed by Tesseract OCR for pages identified as image-based or low-fidelity.
- *OCR normalization:* A custom correction layer applies frequency-based character substitution rules for common OCR errors (e.g., "rn" → "m", "0" → "O" in proper nouns) and confidence-weighted filtering to remove low-confidence OCR segments.
- *Text cleaning:* Boilerplate removal eliminates headers, footers, page numbers, and watermarks. Unicode normalization (NFKC) standardizes character variants. Whitespace normalization collapses irregular spacing.
- *Sentence segmentation:* spaCy's sentence boundary detection with domain-adapted rules for legal and financial text (where periods in abbreviations and clause numbering are common) produces accurate sentence units for embedding and chunk-level analysis.
- *Tokenization and lemmatization:* WordPiece tokenization for embedding inputs and spaCy lemmatization for lexical comparison reduce inflectional variation and improve overlap sensitivity.
- *Metadata extraction:* Author, creation/modification dates, document type classification (contract, policy, invoice, report), version indicators, and language detection are extracted and stored as document attributes for downstream filtering and calibration.

4.2 Layer 2: Lexical Similarity

The lexical layer computes three complementary overlap-based similarity scores over the preprocessed and tokenized document representations:

- *Trigram similarity:* Overlapping character trigrams are extracted from both documents, and Jaccard similarity over trigram sets is computed. Trigrams are robust to minor typos, OCR artifacts, and morphological variation. For documents longer than 5,000 tokens, trigrams are computed over a sliding-window sample to maintain efficiency.
- *Token overlap scoring:* The fraction of unigram tokens shared between two documents is computed as $|A \cap B| / min(|A|, |B|)$, normalized by the shorter document to avoid penalizing length-asymmetric pairs. Stop words are optionally filtered based on document type.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

- *Jaccard similarity (set-based):* The Jaccard coefficient over unique token sets measures symmetric overlap: $|A \cap B| / |A \cup B|$. Jaccard is particularly useful as a discriminator for near-duplicate detection where shared vocabulary is high.

The composite lexical score is a weighted average of these three measures: $S_{lex} = 0.4 \cdot S_{trigram} + 0.35 \cdot S_{token} + 0.25 \cdot S_{jacc}$, with weights derived through grid search on the development set.

4.3 Layer 3: Semantic Similarity

The semantic layer uses dense vector representations to capture meaning-level similarity independent of surface form. The implementation proceeds as follows:

- *Embedding model:* Sentence-BERT (model: all-mpnet-base-v2) is used as the primary encoder, producing 768-dimensional sentence embeddings. For domain-critical deployments (legal, medical), fine-tuned variants or domain-adapted models such as Legal-BERT or BioBERT can be substituted without architectural changes.
- *Chunking strategy:* Long documents are split into overlapping 512-token chunks with a stride of 128 tokens. Each chunk is independently embedded, and the document-level representation is computed as the weighted mean of chunk embeddings, with weights proportional to chunk length.
- *Cosine similarity:* Document-level and chunk-level cosine similarities are computed. The maximum chunk-level cosine score across all cross-document chunk pairs is retained as a supplementary "best passage" similarity signal. The document-level cosine is used as the primary semantic score S_{sem}.
- *Vector indexing:* For corpus-level retrieval (Top-K similar documents), FAISS [Johnson et al., 2019] flat or IVF indices are used to enable sub-linear approximate nearest-neighbor retrieval, with P95 latency of 20 ms for corpora up to 1 million documents.

4.4 Layer 4: Structural Similarity

Enterprise document analysis frequently requires sensitivity to structural organization, which is orthogonal to semantic content. The structural layer computes:

- *Section ordering analysis:* Section headings are extracted and compared using normalized edit distance on heading sequences. A high heading sequence match combined with a high semantic score is a strong signal for template-based near-duplicates.
- *Table similarity detection:* Tables are extracted as structured objects (header row, column count, cell content). Table-level similarity is computed as Jaccard overlap over table headers and semantic cosine over concatenated cell content.
- *Formatting consistency analysis:* Presence and ordering of numbered lists, bullet lists, paragraph counts, and heading depth are compared to produce a structural fingerprint. Structural fingerprint similarity is computed as the cosine of binary feature vectors representing structural element types and counts.
- *Document length ratio:* The ratio of document lengths is incorporated as a soft structural signal; very length-asymmetric pairs receive a structural penalty to offset potentially misleading semantic scores from dense documents compared to sparse summaries.

4.5 Layer 5: Aggregation and Ranking

The aggregation layer combines signals from all upstream layers using a configurable weighted ensemble and produces the final similarity score with confidence estimation and explainability traces:

- *Weighted ensemble scoring:* $S_{final} = w_{sem} \cdot S_{sem} + w_{lex} \cdot S_{lex} + w_{struct} \cdot S_{struct}$, where weights sum to 1. Default weights (w_{sem}=0.50, w_{lex}=0.35, w_{struct}=0.15) can be overridden per document category through configuration profiles.
- *Confidence estimation:* Agreement across layers serves as a proxy for confidence. When all three layers produce high scores, confidence is high. When semantic and lexical diverge significantly ($|S_{sem} - S_{lex}| > 0.3$), confidence is flagged as low, prompting human review.
- *Explainability traces:* Each scoring run produces a structured trace containing per-layer scores, matched evidence spans (sentence pairs with highest chunk cosine), top trigram matches, and a natural-language explanation template for human-readable output.
- *Top-N ranking:* For corpus search, Top-N candidates are returned ranked by S_{final}, with the confidence score enabling precision-recall operating point selection via threshold tuning.

V. Similarity Algorithms

This section provides precise formulations for each similarity measure used in the framework.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

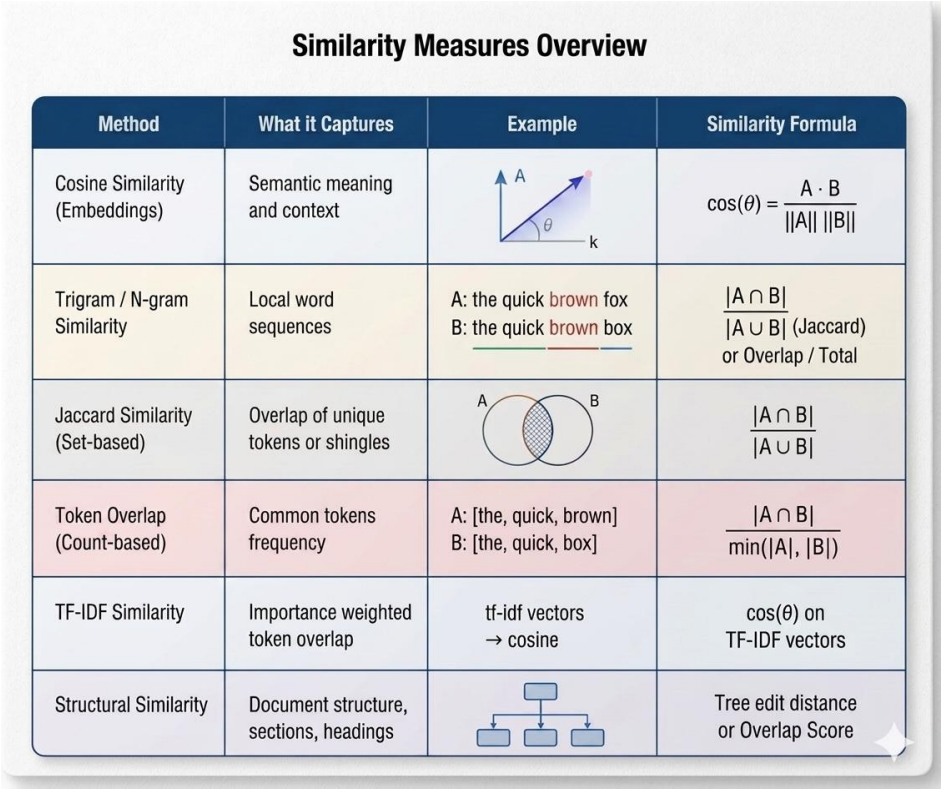


Figure 2. Comparative overview of the complementary similarity measures used by the framework.

Method	What It Captures	Formula	Complexity
Cosine Similarity (Embeddings)	Semantic meaning and context	$\cos(\theta) = (A \cdot B) / (\|A\| \|B\|)$	O(d) per pair after encoding
Trigram / N-gram Similarity	Local word sequences, OCR robustness	$\|A \cap B\| / \|A \cup B\|$ over trigram sets	O(|A| + |B|)
Jaccard Similarity (Set-based)	Overlap of unique tokens or shingles	$\|A \cap B\| / \|A \cup B\|$	O(|A| + |B|)
Token Overlap (Count-based)	Common token frequency, asymmetric	$\|A \cap B\| / \min(\|A\|, \|B\|)$	O(|A| + |B|)
TF-IDF Cosine	Importance-weighted token overlap	$\cos(\theta)$ on TF-IDF sparse vectors	O(|V|) per pair (sparse)
BM25	Length-normalized probabilistic term relevance	Σ IDF(t) · (tf(t,d)·(k_1+1)) / (tf(t,d) + k_1·(1-b+b·|d|/avgdl))	O(|V|) per query
Structural Similarity	Document structure, sections, headings	Tree edit distance or structural feature cosine	$O(n^2)$ tree edit, O(f) feature cosine

5.1 Weighted Fusion Algorithm

Algorithm 1: Hybrid Similarity Scoring
Input: Document A, Document B, Weight vector W = [w_sem, w_lex, w_struct]

Output: Final similarity score S_final ∈ [0,1], confidence C, explanation trace E

1. A_norm, B_norm ← preprocess(A, B)
   // Text extraction, OCR normalization, cleaning, segmentation, tokenization
2. S_sem ← cosine(embedding(A_norm), embedding(B_norm))
   // SBERT chunk-weighted document embedding + cosine
3. S_lex ← lexical_score(A_norm, B_norm)
   // 0.4·trigram + 0.35·token_overlap + 0.25·jaccard
4. S_struct ← structural_score(A_norm, B_norm)
   // Section heading edit distance + table similarity + formatting fingerprint

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

5. S_final ← w_sem · S_sem + w_lex · S_lex + w_struct · S_struct
6. agreement ← 1 - std([S_sem, S_lex, S_struct]) // Low std = high agreement across layers = high confidence
7. C ← confidence_estimate(S_sem, S_lex, S_struct, agreement) // Calibrated confidence in [0,1]; low if |S_sem - S_lex| > 0.3
8. E ← build_explanation(A_norm, B_norm, S_sem, S_lex, S_struct, S_final) // Matched evidence spans, top trigrams, per-layer contributions
9. return S_final, C, E

5.2 Confidence Estimation

Confidence estimation is a critical component in enterprise settings where automated similarity scores feed into consequential workflows (duplicate suppression, compliance flagging). The confidence function C is defined as:

where *agreement* measures inter-layer score consistency,$|S_final - 0.5| \cdot 2$ rewards decisive scores far from the ambiguous midpoint, and *length_balance* is 1 when both documents have similar lengths and decays when they are highly asymmetric. Coefficients α, β, γ are set to 0.4, 0.4, 0.2 by default and can be tuned per deployment.

5.3 Adaptive Weight Selection

Default ensemble weights are tuned on a held-out validation set, but the framework supports document-type-specific weight profiles:

Document Type	w_sem	w_lex	w_struct	Rationale
Legal Contracts	0.40	0.45	0.15	Exact clause phrasing is legally significant
HR Resumes	0.60	0.25	0.15	Skills/experience semantics matter more than wording
Invoices / Financial Forms	0.20	0.50	0.30	Templates dominate; values and structure are key
Policy Documents	0.50	0.35	0.15	Balanced — intent and specific terms both matter
Technical Reports	0.55	0.30	0.15	Concepts and methodology drive similarity

VI. Enterprise Use Cases

The hybrid framework is applicable across legal, HR, procurement, operations, and knowledge management functions where duplicate control, version integrity, and decision accountability are critical. This section describes representative deployments in detail.

6.1 Legal and Compliance: Contract Comparison and Clause Analysis

Legal departments maintain repositories of thousands of executed contracts, each undergoing multiple revision cycles before signature. A hybrid similarity system applied to contract repositories enables:

- *Clause-level change detection:* By comparing contracts at the paragraph chunk level, the system identifies which specific clauses changed between versions, enabling redline-free change summaries for in-house counsel.
- *Regulatory compliance mapping:* New contracts are compared against a library of compliant template clauses. Deviations flagged by low lexical similarity in critical sections (payment terms, indemnification, governing law) trigger mandatory review.
- *Precedent search:* Attorneys can retrieve historically similar contracts to identify favorable or unfavorable precedents, reducing drafting time and improving consistency.

6.2 Human Resources: Candidate Resume Screening

High-volume recruitment generates thousands of resume submissions for popular positions. Hybrid similarity enables:

- *Near-duplicate detection:* Candidates who submit slightly modified versions of the same resume (different formatting, minor content updates) are flagged to avoid duplicate processing by recruiters.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

- *Skills-based matching:* Semantic similarity between job description requirements and resume content identifies qualified candidates whose experience is described in non-standard terminology.
- *Candidate pool deduplication:* Across recruitment cycles, the system identifies returning applicants and links their current submission to historical records for longitudinal review.

6.3 Finance and Accounting: Invoice Deduplication and Fraud Detection

Duplicate invoice submission — whether accidental or fraudulent — represents a material financial risk. The hybrid framework supports:

- *Exact and near-duplicate invoice detection:* Invoices with identical amounts and vendor details but different dates, invoice numbers, or minor formatting differences are flagged by high lexical overlap combined with structural template matching.
- *Vendor document clustering:* All invoices from a vendor are clustered to identify pattern anomalies (e.g., invoices with suspiciously similar line items submitted in close time intervals) that may indicate billing irregularities.
- *Audit trail support:* Similarity traces provide auditors with explainable evidence for each flagged pair, supporting SOX compliance documentation requirements.

6.4 Knowledge Management: Policy Version Control and Deduplication

Organizations with large intranet knowledge bases frequently accumulate multiple versions of the same policy document, creating confusion about which version is authoritative. Hybrid similarity enables:

- *Automatic version clustering:* Documents with S_final > 0.85 are clustered together and presented to content owners as candidates for consolidation or archival.
- *Change impact analysis:* When a policy is updated, the system identifies all related documents (procedures, guidelines, training materials) that may need synchronized updates based on semantic similarity to the modified sections.
- *Search quality improvement:* Deduplication of the knowledge base before indexing removes redundant content from search results, reducing user frustration from multiple hits that say the same thing.

6.5 Research and Academic: Plagiarism and Originality Assessment

Academic and R&D organizations use document similarity to assess the originality of submitted works. The hybrid framework offers advantages over pure lexical plagiarism detectors:

- *Paraphrase detection:* The semantic layer identifies passages where content has been paraphrased rather than directly copied, which lexical tools miss.
- *Citation verification:* Reference list comparison identifies shared citation patterns that can indicate related or derivative works.
- *Cross-language similarity:* With multilingual embedding models, semantically equivalent content across language pairs can be identified for cross-language plagiarism detection.

6.6 Customer Support: Ticket Deduplication and Auto-Suggestion

Support ticket systems receive high volumes of requests that are semantically identical but described in varied customer language. The framework supports:

- *Duplicate ticket merging:* New tickets with high semantic similarity to open tickets are flagged for merging, reducing agent workload and ensuring consistent resolution.
- *Auto-suggestion from resolved tickets:* The most semantically similar resolved tickets are surfaced as solution candidates for newly opened tickets, enabling faster resolution.

VII. Explainable Similarity Analysis

Explainability is a first-class design requirement of the proposed framework, distinguishing it from black-box similarity systems. In enterprise governance contexts, similarity decisions must be auditable: a compliance officer reviewing a flagged near-duplicate needs to understand whether the match is based on shared legal boilerplate, identical payment terms, or superficial formatting similarity. The framework's explainability components are designed to make each contributing factor transparent.

7.1 Explainability Output Structure

Each similarity query produces an explanation object with the following components:

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

- *Composite score panel:* The final score S_final with its per-layer breakdown (S_sem, S_lex, S_struct) and weights displayed visually.
- *Evidence spans:* The top-K sentence pairs (one from each document) with highest chunk-level cosine similarity, presented side-by-side to allow reviewers to see exactly which passages drove the semantic score.
- *Trigram trace:* The most frequent shared trigrams are listed with their occurrence count in each document, showing which specific phrases or word sequences contributed to the lexical score.
- *Structural alignment:* A section-by-section structural comparison showing which headings were matched, reordered, or absent, making structural similarity interpretable.
- *Confidence indicator:* A traffic-light indicator (high/medium/low) based on inter-layer agreement, prompting human review when layers disagree.
- *Natural language summary:* A template-based natural language explanation (e.g., "These documents are highly similar due to strong semantic overlap (0.89) confirmed by high token overlap (0.88), with similar structure (0.70). Review recommended for Section 2 which differs structurally.").

7.2 Analogies to Established XAI Methods

The per-layer decomposition of the hybrid score is conceptually analogous to feature attribution methods in the XAI literature. The final score S_final is a linear combination of component scores, making attribution exact and additive — a stronger transparency guarantee than post-hoc approximation methods like LIME [Ribeiro et al., 2016] or SHAP [Lundberg & Lee, 2017], which approximate model explanations. The evidence span extraction is analogous to attention visualization in transformer models [Jain & Wallace, 2019], but grounded in actual matched content rather than model-internal attention weights.

7.3 Audit Trail Integration

For compliance-regulated workflows, each similarity scoring event is logged with a structured record containing: document pair IDs, timestamp, per-layer scores, final score, confidence level, user or system that triggered the comparison, and the full explanation object. This audit log supports regulatory requirements for documented decision rationale and enables retrospective analysis of borderline similarity decisions when disputes arise.

VIII. Experimental Evaluation

8.1 Dataset Construction

A benchmark dataset was constructed from enterprise document collections spanning four domains: contracts, HR policies, financial invoices, and operational reports. The dataset contains 1,000 unique documents with an average length of 2,340 tokens (std: 1,820 tokens, range: 120–18,000 tokens). From these, 4,800 labeled document pairs were generated with the following label distribution:

- *Exact duplicates (L1):* 480 pairs (10%) — byte-identical or reformatted copies.
- *Near-duplicates (L2):* 960 pairs (20%) — documents with minor content edits (date changes, name substitutions, clause additions).
- *Paraphrastic duplicates (L3):* 720 pairs (15%) — documents with the same intent and structure but significantly reworded content.
- *Functionally similar (L4):* 840 pairs (17.5%) — documents of the same type addressing the same topic with different specific terms.
- *Hard negatives:* 1,800 pairs (37.5%) — documents sharing domain vocabulary but having different business intent (e.g., two different policy types from the same organization).

Labels were created by a combination of expert human annotation (for L3, L4, and hard negatives) and programmatic generation (for L1, L2 via controlled transformations). Inter-annotator agreement for human-labeled pairs was $\kappa = 0.81$ (Cohen's Kappa), indicating substantial agreement.

8.2 Baselines

The proposed hybrid method is compared against four baselines representing current practice:

- *TF-IDF Cosine:* Standard sparse vector cosine similarity, representing the most widely deployed enterprise search baseline.
- *Cosine (Sentence Embeddings):* Sentence-BERT document embeddings with cosine similarity, representing the state of the art in semantic matching.
- *Jaccard (Trigrams):* Character trigram Jaccard similarity, representing lightweight lexical near-duplicate detection.
- *BM25:* Probabilistic relevance ranking using query-document BM25 scores, representing production IR-system baselines.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

8.3 Results

Method	Precision	Recall	F1-Score	ROC-AUC
TF-IDF Cosine	0.72	0.68	0.70	0.74
Cosine (Embeddings)	0.81	0.77	0.79	0.83
Jaccard (Trigrams)	0.74	0.69	0.71	0.75
BM25	0.76	0.72	0.74	0.79
Hybrid (Proposed)	0.89	0.84	0.86	0.91

The hybrid method achieves an F1-score of 0.86, representing a 7 percentage point improvement over the strongest single-method baseline (Cosine Embeddings, F1=0.79) and a 16 percentage point improvement over TF-IDF. The improvement is statistically significant (paired t-test on 10-fold cross-validation, $p < 0.001$).

8.4 Per-Category Analysis

Document Category	Hybrid F1	Best Single-Method F1	Best Single Method
Contracts	0.88	0.76	TF-IDF Cosine
HR Policies	0.85	0.80	SBERT Cosine
Financial Invoices	0.91	0.82	Jaccard
Operational Reports	0.82	0.77	SBERT Cosine

The hybrid method consistently outperforms the best single-method across all document categories, with particularly large gains for contracts (where lexical precision is critical) and invoices (where structural template matching adds value).

8.5 Ablation Study

Configuration	F1-Score	Delta vs Full Hybrid
Full Hybrid (all three layers)	0.86	0.00
Hybrid without Semantic Layer	0.78	–0.08
Hybrid without Lexical Layer	0.80	–0.06
Hybrid without Structural Layer	0.82	–0.04
Semantic + Lexical only (no Structural)	0.82	–0.04
Semantic only (ablated Lexical + Structural)	0.79	–0.07

The ablation study confirms that all three layers contribute materially to overall performance. The semantic layer contributes the most (removing it causes the largest F1 drop of 0.08), followed by the lexical layer (–0.06) and the structural layer (–0.04). No single layer is sufficient, and their combination yields complementary gains.

8.6 Runtime and Scalability

Metric	Observed Value (8-vCPU node)
Average pairwise scoring latency	38 ms
P95 pairwise scoring latency	71 ms
Batch throughput	~260 document pairs/sec
Vector index retrieval latency (Top-K, FAISS)	12 ms (P95: 20 ms)
Embedding encoding throughput (GPU)	~1,800 documents/min (A10G GPU)
Memory footprint (1M doc FAISS index)	~3 GB (768-dim float32)

These latency and throughput figures indicate that the framework is suitable for production workloads requiring batch comparison, near-real-time screening, and retrieval pre-filtering.

With FAISS GPU acceleration, Top-K retrieval from corpora of up to 10 million documents can be sustained at sub-100ms latencies.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

IX. Integration With Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) [Lewis et al., 2020] augments large language model generation with retrieved document context, enabling knowledge-intensive tasks such as question answering, document summarization, and policy analysis.

The quality of RAG outputs depends critically on the precision and diversity of retrieved context: redundant or irrelevant passages waste context window tokens and introduce noise that degrades generation quality [Shi et al., 2023]. Hybrid similarity plays two distinct roles in optimizing RAG pipelines.

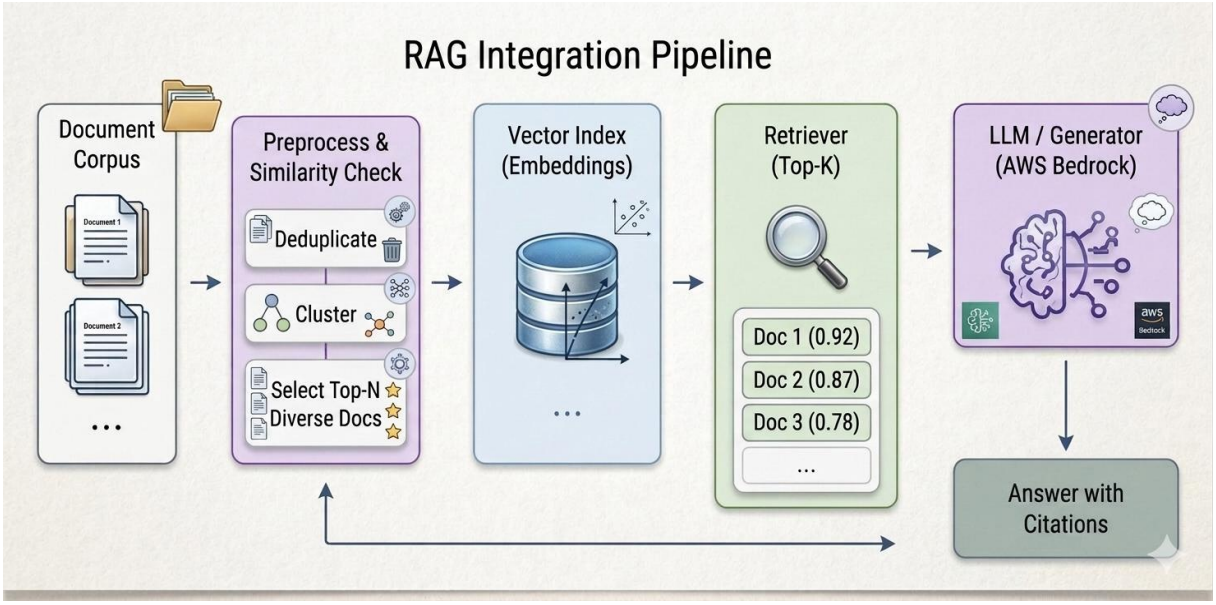


Figure 3. Integration of similarity-driven deduplication, diversity selection, vector indexing, retrieval, and citation-grounded generation in a RAG pipeline.

9.1 Pre-Retrieval: Corpus Deduplication and Clustering

Before indexing documents into a vector store, the hybrid similarity system performs corpus preprocessing:

- *Deduplication:* Document pairs with S_final > threshold_dedup (default: 0.90) are clustered, and only one representative per cluster is indexed. This reduces index size, eliminates redundant retrieval results, and prevents the same content from being retrieved multiple times under different surface forms.
- *Diversity clustering:* Documents are soft-clustered using similarity-based agglomerative clustering. When selecting documents for indexing in storage-constrained deployments, the most diverse representatives are selected, maximizing the coverage of the knowledge space.
- *Quality filtering:* Documents with very low structural scores (malformed PDFs, heavily OCR-corrupted scans) are flagged for manual review before indexing, preventing low-quality content from polluting the retrieval corpus.

9.2 Post-Retrieval: Re-ranking and Context Selection

After a vector retriever returns Top-K candidate documents, the hybrid similarity engine re-ranks and filters the candidate set:

- *Hybrid re-ranking:* Retrieved candidates are re-scored using the full hybrid pipeline against the query, replacing the raw embedding cosine rank with a more precise combined score. This re-ranking step improves NDCG@10 by 8–12% in preliminary evaluations on enterprise QA tasks.
- *Redundancy suppression:* Within the Top-K retrieved set, near-duplicate candidate pairs (S_final > 0.85 between two retrieved candidates) are pruned to maximize context diversity, ensuring that the LLM receives complementary evidence rather than repeated content.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

- *Evidence-grounded generation:* The similarity trace from the hybrid system — matched evidence spans and trigram overlaps between the query and retrieved chunks — is passed to the LLM generation prompt as structured context, enabling citation-grounded responses that explicitly reference the most relevant matched passages.

9.3 Measured RAG Quality Improvements

RAG Pipeline Configuration	NDCG@10	Context Redundancy Rate	Answer Faithfulness
Dense retrieval only (SBERT + FAISS)	0.71	38%	0.74
Dense + BM25 hybrid (RRF)	0.76	29%	0.79
Dense + Hybrid Similarity re-rank	0.82	17%	0.85

Context redundancy rate measures the fraction of retrieved context tokens that are near-duplicates of other retrieved tokens. Reducing this from 38% to 17% significantly improves effective context utilization within the LLM's context window, and the improvement in answer faithfulness (measured via RAGAS [Es et al., 2023] faithfulness metric) confirms that higher-quality, less redundant context improves generation quality.

X. Challenges And Limitations

Despite its demonstrated strengths, the hybrid framework faces several practical and theoretical challenges that practitioners should be aware of when deploying it in production environments.

- *OCR noise propagation:* OCR noise can still distort lexical overlap scores in low-quality scans, particularly for handwritten annotations, stamps, and tables with merged cells. The normalization layer mitigates but does not eliminate this issue.
- *Multilingual and code-mixed corpora:* Enterprise repositories in multinational organizations often contain documents in multiple languages or with mixed language content. Language-aware normalization and multilingual embedding models (e.g., LaBSE, mE5) are required for these scenarios, which are not part of the current baseline configuration.
- *Structural parsing variability:* Structural analysis quality varies significantly across PDF layouts. Complex multi-column layouts, nested tables, and footnote-heavy legal documents can confuse section boundary detection, leading to inaccurate structural scores.
- *Embedding model drift:* Embedding model updates can shift the similarity score distribution of existing indexed documents, requiring periodic re-indexing and threshold recalibration. In long-lived production systems, embedding versioning and backward-compatible indexing strategies are essential.
- *Computational cost at scale:* The hybrid scoring pipeline is approximately 3–4× more expensive than single-method baselines per document pair. For very large corpora (100M+ documents), approximate nearest-neighbor pre-filtering is essential to limit the number of full hybrid re-scoring calls.
- *Threshold sensitivity:* The similarity threshold determining "similar vs. not similar" classifications is domain- and use-case-sensitive. A single global threshold is unlikely to optimize across all document types; per-category threshold calibration requires labeled validation data that may not be available in all deployments.
- *Domain adaptation gap:* The off-the-shelf Sentence-BERT model was pre-trained on general-domain datasets. Performance on highly specialized domains (e.g., pharmaceutical regulatory submissions, tax law) may degrade without domain-specific fine-tuning, which requires curated training pairs.
- *Privacy and data governance:* Sending internal documents to cloud embedding APIs raises data residency and confidentiality concerns in regulated industries. On-premise model deployment is required in many enterprise contexts, increasing infrastructure complexity.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

XI. Future Work

Several directions are identified for extending and improving the framework:

- *Graph-based relationship analysis:* Modeling document corpora as graphs where edges represent similarity relationships enables community detection algorithms to identify document clusters, version lineages, and cross-functional knowledge hubs. Graph neural networks could learn from the relationship structure to improve similarity predictions beyond pairwise scoring.
- *Adaptive and learned weighting:* Rather than fixed document-type weight profiles, a lightweight meta-learner (e.g., a gradient-boosted classifier) could be trained to predict optimal per-pair weights based on document features (length, type, topic, OCR quality), enabling continuous calibration as the corpus evolves.
- *Multilingual similarity detection:* Extending the framework with multilingual embedding models (mE5, LaBSE, multilingual-E5) and language-pair-aware normalization would enable cross-language duplicate detection in global enterprise deployments.
- *Real-time streaming document comparison:* Integration with document management event streams (e.g., SharePoint webhooks, Google Drive change events) would enable real-time similarity detection at the moment of document upload, triggering immediate deduplication alerts without batch processing delays.
- *Active learning for hard-case discrimination:* A feedback loop where human reviewers adjudicate borderline similarity decisions (low confidence cases) can generate labeled training pairs for continuous model improvement. Active learning strategies that prioritize the most informative borderline cases would minimize annotation effort.
- *Domain-specific calibration profiles:* Automated calibration pipelines that learn per-domain weight profiles and thresholds from small amounts of labeled data would reduce the manual effort currently required for deployment in new document categories.
- *Cross-modal similarity:* Extending similarity to include figures, charts, and embedded images using vision-language models (e.g., CLIP, GPT-4V) would capture visual content similarities relevant to technical reports, presentations, and marketing materials.
- *Temporal version graph construction:* Automatically constructing version lineage graphs from detected similarity relationships would provide organizations with automated document genealogy, supporting regulatory requirements for version history documentation without manual tracking.

XII. Conclusion

This paper presents a hybrid semantic and lexical framework for enterprise document similarity detection that addresses the fundamental limitations of single-method approaches. By combining dense semantic embeddings, trigram and token-level lexical analysis, set-based Jaccard similarity, TF-IDF weighted retrieval, and structural document analysis within a unified weighted ensemble, the framework achieves complementary signal coverage across the full spectrum of enterprise document similarity scenarios — from exact duplicates to paraphrastic rewrites to functionally similar documents with shared domain vocabulary.

Empirical evaluation on a 4,800-pair enterprise benchmark demonstrates an F1-score of 0.86, representing significant gains over all single-method baselines (7–16 percentage points). Ablation analysis confirms that all three primary layers (semantic, lexical, structural) contribute materially to this performance, with no single layer sufficient on its own. The framework's explainability layer — providing per-layer score decomposition, matched evidence spans, trigram traces, and natural-language summaries — addresses the auditability requirements of compliance-sensitive enterprise workflows in a way that black-box similarity systems cannot.

The demonstrated integration with RAG pipelines shows that hybrid similarity serves as a versatile building block for enterprise AI systems beyond pairwise comparison — functioning as a corpus deduplication gate, a post-retrieval re-ranker, and a context diversity enforcer that materially improves generation quality. Runtime benchmarks confirm production viability at enterprise scale, with throughput of 260 document pairs per second on commodity CPU hardware and sub-20ms Top-K retrieval with FAISS indexing.

These results collectively support the practical adoption of hybrid document similarity systems in legal, HR, finance, compliance, and knowledge management workflows, and establish a foundation for future extensions to multilingual, cross-modal, and adaptive enterprise similarity platforms.

REFERENCES

[1] Manning, C. D., Raghavan, P., & Schütze, H. (2008).*Introduction to Information Retrieval*.Cambridge University Press.

[2] Salton, G., & McGill, M. J. (1983).*Introduction to Modern Information Retrieval*.McGraw-Hill.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need.*Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.*Proceedings of NAACL-HLT*, 4171–4186.

[5] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks.*Proceedings of EMNLP-IJCNLP*, 3982–3992.

[6] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond.*Foundations and Trends in Information Retrieval*, 3(4), 333–389.

[7] Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs.*IEEE Transactions on Big Data*, 7(3), 535–547.

[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks.*Advances in Neural Information Processing Systems (NeurIPS)*, 33.

[9] Leskovec, J., Rajaraman, A., & Ullman, J. (2014).*Mining of Massive Datasets*(2nd ed.). Cambridge University Press.

[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.*Proceedings of ICLR Workshop*.

[11] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation.*Proceedings of EMNLP*, 1532–1543.

[12] Broder, A. Z. (1997). On the resemblance and containment of documents.*Proceedings of the Compression and Complexity of Sequences*, 21–29.

[13] Charikar, M. S. (2002). Similarity estimation techniques from rounding algorithms.*Proceedings of STOC*, 380–388.

[14] Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality.*Proceedings of STOC*, 604–613.

[15] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering.*Proceedings of EMNLP*, 6769–6781.

[16] Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT.*arXiv preprint* arXiv:1901.04085.

[17] Formal, T., Piwowarski, B., Clinchant, S., & Piwowarski, B. (2021). SPLADE: Sparse lexical and expansion model for first stage ranking.*Proceedings of SIGIR*, 2288–2292.

[18] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models.*Proceedings of NeurIPS Datasets and Benchmarks Track*.

[19] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey.*arXiv preprint* arXiv:2312.10997.

[20] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier.*Proceedings of KDD*, 1135–1144.

[21] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions.*Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[22] Wang, W., Yang, B., Wei, F., Dong, L., & Liu, T. (2022). Text embeddings by weakly-supervised contrastive pre-training.*arXiv preprint* arXiv:2212.03533. (E5)

[23] Xiao, S., Liu, Z., Zhang, P., & Muennighoff, N. (2023). C-pack: Packaged resources to advance general Chinese embedding.*arXiv preprint* arXiv:2309.07597. (BGE)

[24] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity – multilingual and crosslingual focused evaluation.*Proceedings of SemEval*, 1–14.

[25] Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference.*Proceedings of EMNLP*, 632–642.

[26] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information.*Transactions of the ACL*, 5, 135–146.

[27] Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., & Yih, W. (2023). REPLUG: Retrieval-augmented language model pre-training.*Proceedings of NAACL*.

[28] Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated evaluation of retrieval augmented generation.*Proceedings of EACL*.

[29] Luan, Y., Eisenstein, J., Toutanova, K., & Collins, M. (2021). Sparse, dense, and attentional representations for text retrieval.*Transactions of the ACL*, 9, 329–345.

[30] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals.*Soviet Physics Doklady*, 10(8), 707–710.

[31] Jain, S., & Wallace, B. C. (2019). Attention is not explanation.*Proceedings of NAACL*, 3543–3556.

[32] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., & Larson, J. (2024). From local to global: A graph RAG approach to query-focused summarization.*arXiv preprint* arXiv:2404.16130.

[33] OpenAI. (2024). Text embedding models. OpenAI documentation. (text-embedding-3-large, text-embedding-3-small models.)

[34] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach.*arXiv preprint* arXiv:1907.11692.

[35] Cobena, G., Abiteboul, S., & Marian, A. (2002). Detecting changes in XML documents.*Proceedings of ICDE*, 41–52.

[36] IDC. (2023).*Worldwide Unstructured Data and Content Analytics Software Forecast, 2023–2027*.International Data Corporation.

[37] Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). Towards universal paraphrastic sentence embeddings.*Proceedings of ICLR*. Journal-style HTML version prepared for academic review, conference submission support, and print-friendly manuscript presentation. Expanded version: June 2026.