International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered Multilingual Plagiarism Detection Using Graph Neural Networks and Syntax-Semantic Knowledge Graphs: A Review

Anuja Wagh[1], Rutuja Vanwe[1], Gargi Madane[1], Prof. Dhage S. A.[2]
[1]Student, Department of Computer Engineering
[2]Assistant Professor, Department of Computer Engineering Vishwabharti Academy's College of Engineering, Ahilyanagar (MH), India
Savitribai Phule Pune University, Pune (MH), India

***Abstract*—Digital content, machine translation and generative AI tools make it easy to reuse an idea without reusing its wording, and this has made plagiarism harder to detect in academic and research settings. Detectors built on lexical matching were designed for verbatim copying and lose most of their signal once a text is translated, paraphrased or restructured. Knowledge graphs (KGs) and graph neural networks (GNNs) offer an alternative, because they encode relations between words and concepts instead of surface strings. This paper reviews that line of work and concentrates on the CL-UniDepWord-GNN framework of Bouaine et al. published in IEEE Access, which merges WordNet semantic relations with Universal Dependencies syntax in a single multilingual graph and learns over it with relation embeddings, a graph convolutional network (GCN) and a graph attention network (GAT). We summarise how the framework is built, which datasets it uses and what PlagDet scores it reports for four language pairs, and we read those results critically, including how the evaluation corpora were assembled and how the baselines were compared. We then collect the open problems, namely evaluation realism, dependence on linguistic resources, computational cost, explainability and AI-generated text, and outline a prototype architecture for future work.**

***Index Terms*—Plagiarism Detection, Graph Neural Networks, Knowledge Graphs, Cross-Language Plagiarism Detection, Natural Language Processing, WordNet, Universal Dependencies, Graph Attention Networks, Graph Convolutional Networks, Semantic Similarity**

I. Introduction

Text that was once copied from a printed page now moves through search engines, machine translation and text generators, and the boundary between borrowing and original work is harder to police than it used to be. Verbatim copying is only one case. A passage can be translated into another language, rewritten sentence by sentence, assembled from several sources, or regenerated by a language model that keeps the argument and discards the wording. Detectors that compare character or word sequences were designed for the first case and have little to hold on to in the others.

Cross-language plagiarism is the clearest example. When a text is translated and presented as new, the intellectual content is unchanged but almost no surface token survives, so lexical overlap with the source is close to zero.

Two developments motivate the work examined here. First, representations that encode meaning, namely multilingual embeddings [6], [24], knowledge graphs (KGs) [22] and graph neural networks (GNNs) [25], allow two texts to be compared by what they say rather than by how they are spelled. Second, Transformer-based language models [20], [21], [23] can restate a passage without sharing vocabulary with its source, which pushes detection further towards semantic comparison.

This review follows the progression from lexical to semantic to graph-based detection and gives most attention to the CL-UniDepWord-GNN framework of Bouaine et al. [1], which pairs WordNet relations with Universal Dependencies syntax. We describe how the framework is built, restate what it reports, and examine how far the reported numbers support the conclusions drawn from them.

II. Types of Plagiarism

Plagiarism takes several forms:

1) Direct Plagiarism: verbatim reuse without attribution.
2) Paraphrasing Plagiarism: the same content with reworded sentences.
3) Mosaic Plagiarism: fragments from several sources stitched together, often with light edits.
4) Self-Plagiarism: undeclared reuse of one's own earlier published work.
5) Translation Plagiarism: content translated from another language and presented as original.
6) Semantic Plagiarism: reuse of ideas or arguments with little shared wording.
7) AI-Generated Content Plagiarism: machine-written text that restates existing sources.

The first four forms usually leave recognisable stretches of shared text, so lexical comparison can still catch many of them. The last three are different in kind: the evidence of reuse lies in shared meaning, not shared words, and this is why they drive the interest in semantic and graph-based representations.

III. Evolution of Plagiarism Detection Techniques

Early detectors compared texts as strings:

- N-Gram Matching: counting shared word or character sequences.
- String Matching: locating long common substrings.
- Fingerprinting: hashing selected sequences so that documents can be compared compactly.
- TF-IDF Similarity: weighting terms by rarity and comparing document vectors.

These techniques measure overlap in form. Later systems added representations that capture meaning:

- Word Embeddings: dense vectors learned from word co-occurrence.
- Sentence Embeddings: fixed-length vectors for whole sentences.
- Semantic Similarity Analysis: scoring relatedness beyond identical words.
- Knowledge Graph Representations: texts expressed as typed relations between concepts.
- Graph Neural Networks: models that learn over those relations.

Each step trades simplicity and speed for the ability to recognise reuse that has been disguised by translation or rewording.

IV. Literature Review

Work on plagiarism detection has moved from string comparison towards semantic and graph-based learning. The sections below group representative methods by the kind of evidence they use, and Table I summarises them.

A. Lexical Similarity Based Methods

N-gram matching, fingerprinting and TF-IDF vectors remain attractive because they are fast, transparent and easy to explain to a reader: a flagged passage can be shown next to its source with the shared sequences highlighted. They work well when text has been copied. Their weakness is structural. Synonym substitution, reordering and above all translation remove the shared tokens on which they depend, and preprocessing steps such as stemming recover only a small part of what is lost.

B. Semantic Similarity Based Methods

Embedding-based methods compare texts in a vector space instead of at the token level. Sentence-BERT [6] fine-tunes a BERT encoder in a siamese setup so that the cosine similarity of two sentence vectors reflects their meaning, and multilingual encoders [24] extend this across languages. Earlier knowledge-based measures such as the one by Mihalcea et al. [19] combined corpus statistics with WordNet distances. The limitation common to these approaches is that a sentence is reduced to one vector: relations between the concepts inside it are not represented explicitly, and it is difficult to say which parts of two texts actually matched.

C. Graph-Based Multilingual Approaches

Graph-based detectors represent a text as typed relations between concepts and compare texts in that space. Because synsets and dependency labels are shared across languages, the same representation can in principle be built for both sides of a translation. Franco-Salvador et al. built graphs from BabelNet [15] for cross-language detection [14], and their later work paired a knowledge graph with TF-IDF vectors [27]. Stegmu¨ller et al. compared documents through entity vectors drawn from Wikidata [26]. Hourrane and Benlahmar applied a graph transformer to WordNet-based graphs [2]. Bouaine et al. [1] then added Universal Dependencies syntax and learned over the fused graph with RotatE-style relation embeddings [8], a GCN [9] and a GAT [10]. General surveys of graph learning are given in [22], [25].

A caution applies to Table I. The PlagDet values are quoted as given in [1]; they come from different corpora and experimental setups and should not be read as a like-for-like ranking.

TABLE I: Comparative Review of Existing Plagiarism Detection Approaches

Method	Strength	Limitation	Dataset (reported PlagDet)
Lexical (n-gram, TF-IDF)	Fast and transparent	Blind to translation and paraphrase	No single benchmark
Sentence-BERT [6]	Dense semantic vectors; multilingual variants	No explicit relational structure; hard to explain matches	NLI training, STS evaluation in [6]
CL-KGA [14]	BabelNet concepts bridge languages	Depends on BabelNet coverage and sense disambiguation	PAN-PC-10/11; 66.3 (En-Es), 59.5 (En-De)
CL-OSA [26]	Entity-level comparison via Wikidata	Entity vectors discard sentence structure	PAN-PC-11; 57.3 (En-Es), 52.1 (En-De)
Graph Transformer (CL-GTA) [2]	Attention over WordNet-based graphs	No syntactic information, as noted in [1]	PAN-PC-11, JRC-Acquis, Europarl, Wikipedia, conference papers; 62.0 (En-Es), 58.4 (En-Fr), 52.2 (En-Ar)
CL-UniDepWord-GNN [1]	Semantics and syntax in one graph	Relies on WordNet, UD and parser quality; evaluation design (Sec. XII)	PAN11, JRC-Acquis, Europarl, Wikipedia, OPUS, conference papers; 98.50 (En-Es), 89.02 (En-De), 77.67 (En-Fr), 77.65 (En-Ar)

V. Architecture of Traditional Plagiarism Detection Systems

Conventional systems are organised as a lexical pipeline, shown in Fig. 1. A submitted document is normalised (tokenisation, stop-word removal, stemming), compared with a reference collection by n-gram matching or TF-IDF similarity, converted into a similarity score and percentage, and summarised in a report. Every stage after normalisation depends on shared tokens, so a translated or heavily reworded text produces little or no match even when its content is identical to a source.



Fig. 1: Traditional Plagiarism Detection Architecture

Fig. 1 depicts this pipeline, including the two comparison branches (n-gram matching and TF-IDF similarity) that feed the score calculation.

VI. Review of the Base Paper

The base paper [1], published in IEEE Access, proposes CL-UniDepWord-GNN for cross-language plagiarism detection over four pairs: English with Spanish, French, German and Arabic. Its building blocks are:

- WordNet Semantic Relations [7]
- Universal Dependencies [17]
- Knowledge Graph Construction
- RotatE Relation Embeddings [8]
- Graph Convolutional Networks [9]
- Graph Attention Networks [10]

The authors claim three contributions: a hybrid graph that combines lexical-semantic and syntactic relations for multiple language pairs, a layered embedding model, and an ablation that compares WordNet-only, UD-only and hybrid graphs. The framework is evaluated on PAN11, Europarl, JRC-Acquis, Wikipedia, OPUS and a curated set of conference papers, using precision, recall, granularity and PlagDet.

VII. Knowledge Graph Construction

Graph construction in [1] starts from preprocessed sentences (segmentation, noise removal, stop-word removal and lower-casing) and builds two layers over the same set of word nodes, as outlined in Fig. 2.

Semantic layer. Each token is reduced to a (lemma, part-of-speech) pair and looked up in WordNet. Edges are not drawn between all pairs of words. Only tokens inside a five-word window are compared, an edge is created when a synset of one token is linked to a synset of the other by synonymy, antonymy, hypernymy, hyponymy, meronymy or holonymy, and at most three edges are kept per word. Open Multilingual WordNet supplies the non-English lexicons, with OdeNet for German and Arabic WordNet for Arabic; the idea of aligning wordnets across languages goes back to EuroWordNet [12].

Syntactic layer. Each parsed sentence is turned into (head, dependent, relation) triples. spaCy models parse English, Spanish, German and French, and Stanza [11] parses Arabic. Labels such as subject and object come from the shared Universal Dependencies inventory [17], which is what allows the same relation types to be used across languages.

Fusion. Both layers use the same node set, so the hybrid graph is simply the union of their edge sets,

$$G_h = (V,\ E_s \cup E_t), \tag{1}$$

where E_s holds the WordNet edges and E_t the dependency edges, each tagged with its relation type, document identifier and label.

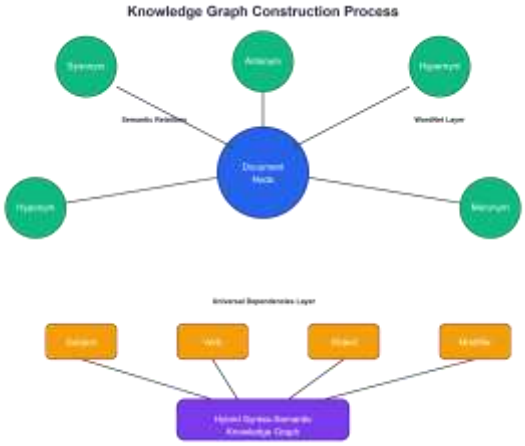


Fig. 2: Knowledge Graph Construction Process

Fig. 2 shows the two layers, WordNet relations (synonym, antonym, hypernym, hyponym, meronym) and dependency roles (subject, verb, object, modifier), joined into one hybrid graph, so that conceptual and grammatical information are available to the model at the same time.

Two design choices deserve comment. The window size and the cap of three edges per word keep the graph small, but the published text justifies them on speed and readability grounds and does not report how results change when they are varied. In addition, no word-sense disambiguation step is described, so an edge can arise through a sense of a word that is not the one intended in context.

VIII. Graph Neural Network Architecture

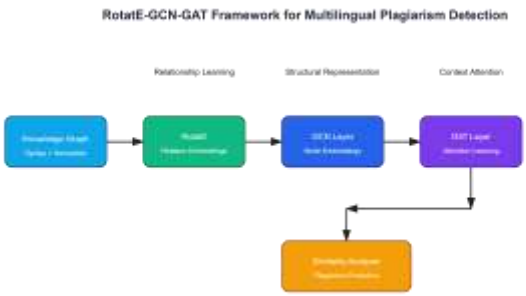


Fig. 3: RotatE-GCN-GAT Architecture

Fig. 3 gives the learning stack of [1]. Nodes start as learned 64-dimensional embeddings and pass through GCN layers [9] that mix each node's representation with those of its neighbours. A multi-head GAT layer [10] then re-weights neighbours by learned attention. Relation types are represented by learnable vectors that the authors describe as inspired by RotatE [8]. For a candidate node pair, the two GAT outputs and the relation vector are concatenated and passed to a classifier,

$$z_{ij} = [\, h_i^{GAT} \mid r_{ij} \mid h_j^{GAT} \,], \qquad \hat{y} = \sigma(\mathrm{MLP}(z_{ij})), \tag{2}$$

with dropout before an MLP and a sigmoid output. Training uses binary cross-entropy, the Adam optimiser with a learning rate of 0.001, batches of 32, ten epochs and dropout of 0.3. The decision threshold was searched between 50% and 70% on validation data and fixed at 63%.

Reading the description closely raises three points. First, a table of learnable relation vectors does not by itself implement RotatE, which models each relation as a rotation in complex space and scores triples accordingly [8]; the training objective for such a scoring function is not given, so "RotatE" is best read as the inspiration for the relation embeddings. Second, the order of the components is not entirely consistent: the overview figure in [1] places relation and node embedding before the GAT, whereas the text applies the GAT to GCN outputs and adds relation vectors afterwards by concatenation. Third, the prediction target is the presence of a labelled link between a node pair, and the text does not explain how pair-level outputs are turned into the passage-level detections that PlagDet is defined on.

IX. Workflow of Multilingual Plagiarism Detection

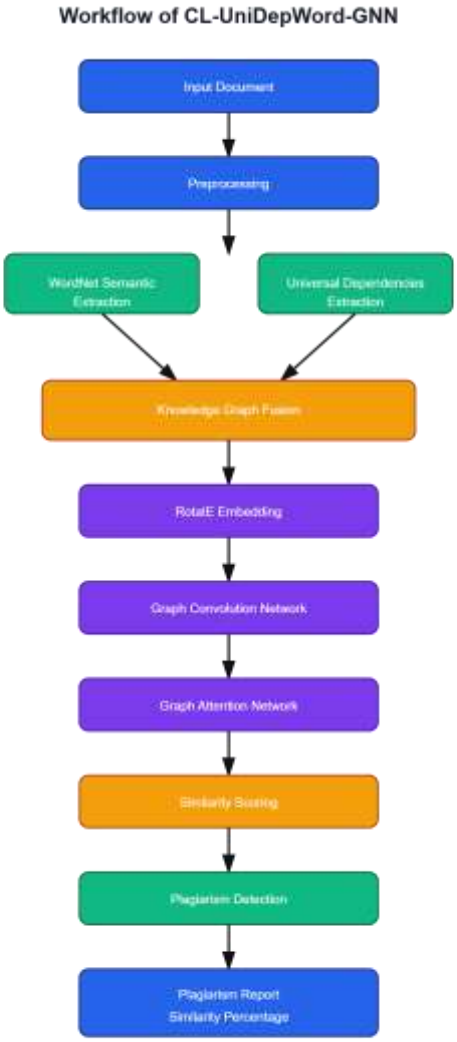


Fig. 4: Workflow of CL-UniDepWord-GNN

Fig. 4 lays out the framework as a sequence: input documents are preprocessed, semantic and syntactic relations are extracted in parallel, the two are fused into one graph, relation and node embeddings are learned by RotatE-style embedding, GCN and GAT, and a final stage produces a similarity score and a plagiarism decision with a percentage in the report. The figure presents the learning stages in a straight line for clarity. In [1] the "similarity scoring" step is the MLP classifier of Eq. (2) applied at a fixed threshold, not a separate cosine comparison.

X. Datasets Used in Recent Research

Table II lists the corpora used in [1] and the role each one plays there, using the corpus summary in that paper. The benchmark collection is PAN-PC-11 [3]; the others are parallel or comparable collections, namely Europarl [4], JRC-Acquis [5] and OPUS [18].

TABLE II: Dataset Characteristics

Dataset	Type	Usage in [1]
PAN11	Benchmark Corpus	Plagiarism cases: 210 En-Es and 2,222 En-De pairs
JRC-Acquis	Legal Parallel Corpus	Positive pairs: about 10,000 each for En-Fr, En-Es and En-De
Europarl	Parallel Corpus	Negative pairs, together with Wikipedia (9,442 En-Fr; 9,423 En-Es; 9,999 En-De)
Wikipedia	Open Corpus	Negative pairs, together with Europarl
OPUS	Parallel Corpus	En-Ar: 151,136 aligned sentence pairs as positives, plus the same number of random non-aligned pairs as negatives
Conference Papers	Curated Academic Texts	En-Fr positives: 620 pairs

The composition matters for interpretation. Most positive examples come from aligned parallel texts rather than from PAN-PC-11, and the negatives are drawn from different collections (Europarl and Wikipedia), so a classifier could partly learn to tell the source corpus apart. For English–Arabic the task is closer to recognising translated sentence pairs than to detecting plagiarism inside longer documents. These are properties of the experimental design and not defects in the model, but they limit what the reported scores can say about real submissions.

XI. Methodology Analysis

The framework in [1] follows five stages:

1) Data Preprocessing
2) Semantic Relation Extraction using WordNet [7]
3) Syntactic Relation Extraction using Universal Dependencies [17]
4) Knowledge Graph Fusion
5) Graph Neural Network Learning

For each language pair the labelled examples are split at random into 60% training, 20% validation and 20% test data. The training part fits the weights, the validation part selects hyperparameters and the threshold, and the test part is used once for the reported figures. Evaluation follows the PAN protocol of precision, recall, granularity and PlagDet [16]. Three ablated variants are trained for each pair (WordNet-only, UD-only and hybrid), which is the most useful feature of the design: it makes it possible to ask which resource contributes what.

Two limits of the protocol are worth stating. A random split of pairs does not guarantee that the same source document is absent from both training and test data, and the article does not report testing on a corpus that was held out entirely. Both would strengthen the evidence for generalisation.

XII. Performance Analysis

All numbers below are quoted from [1]; we did not re-run the experiments. Table III gathers PlagDet for the three variants of every language pair from Table 4 of that paper. One inconsistency in the source should be noted: its comparison table for English–Arabic repeats the English–French WordNet-only row (PlagDet 67.88) where its main results table gives 66.49, and we use the latter.

TABLE III: PlagDet Scores Reported in the IEEE Study

Language Pair	WordNet-only	UD-only	Hybrid
English-Spanish	66.40	95.31	**98.50**
English-German	**91.71**	89.39	89.02
English-Arabic	66.49	72.16	77.65
English-French	67.88	71.60	77.67

Four observations follow from the table.

The hybrid is not uniformly best. It leads the best single-resource variant by 3.19 points for English–Spanish, 6.07 for English–French and 5.49 for English–Arabic, but for English–German it trails WordNet-only by 2.69 points and UD-only by 0.37.

The more useful resource depends on the language. For English–Spanish, UD-only reaches 95.31 against 66.40 for WordNet-only, so nearly all of the improvement over WordNet comes from syntax. For English–German the order reverses. The authors attribute this to the richer WordNet coverage available for German; the explanation is plausible, but it was not tested, for example by degrading lexical coverage, and should be treated as a hypothesis.

Language family is not an obvious predictor. The two weakest results, English–French and English–Arabic, are almost identical (77.67 and 77.65) although the languages are typologically very different. The English–French hybrid run also has the highest granularity among the hybrid runs (1.14, meaning more fragmented detections) and the widest validation spread (±0.0110). Its training data mixes JRC-Acquis legal texts with conference papers, a difference in genre that could contribute, though the paper does not examine this.

The margins over earlier work are large and need careful reading. For English–Spanish the reported 98.50 compares with 66.3 for the best earlier method. The baseline values in the comparison tables coincide with those originally published for PAN-PC data and listed in Table 1 of [1], even though the text states that baselines were re-implemented, so it is unclear whether all methods were run on the same splits. Combined with the dataset composition discussed in Section X, this argues for confirming the gains with leave-one-corpus-out testing, same-source negatives and baselines run on identical splits.

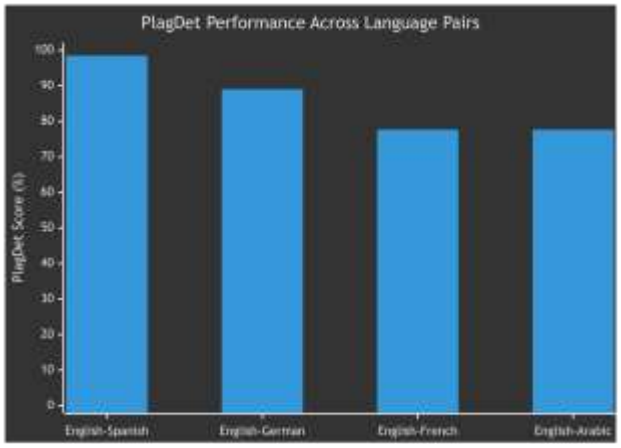


Fig. 5: Performance Comparison Across Language Pairs

Fig. 5 charts the hybrid scores only; Table III adds the two single-resource variants. Taken together, the evidence supports the view that combining semantic and syntactic relations helps in three of the four pairs [1], while the size of the improvement over other published systems remains to be confirmed.

XIII. Comparative Analysis of Existing Approaches

Four formulas recur in the methods reviewed above. Cosine similarity between two document vectors A and B is

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{|A|\,|B|}, \tag{3}$$

and is used by entity- and transformer-based baselines such as CL-OSA [26] and CL-GTA [2]. The framework of [1] does not rely on it and decides with the sigmoid classifier of Eq. (2).

The GCN layer update in the formulation of Kipf and Welling [9] adds self-loops to the adjacency matrix before symmetric normalisation:

$$H^{(l+1)} = \sigma\!\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right), \tag{4}$$

with $\tilde{A} = A + I$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

The GAT normalises attention scores over each node's neighbourhood N_i [10]:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in N_i}\exp(e_{ik})}, \quad e_{ij} = \text{LeakyReLU}\!\left(\mathbf{a}^{\top}[Wh_i \,\|\, Wh_j]\right). \tag{5}$$

Finally, PAN's PlagDet [16] combines the F-measure with granularity,

$$\text{PlagDet} = \frac{F_1}{\log_2(1 + \text{granularity})}, \tag{6}$$

so a detector that reports one case as several fragments is penalised.

Table IV compares capabilities rather than accuracy. In place of the earlier ordinal ratings (Low to Very High), which had no stated criterion, each column records an observable property.

TABLE IV: Comparison of Detection Approaches

Method	Multilingual	Semantic source	Syntax (UD)	Graph Based	AI Detection
TF-IDF	No	None	No	No	No
Sentence-BERT	Via multilingual models	Learned embeddings	No	No	No
Graph Transformer	Yes	WordNet	No	Yes	No
CL-UniDepWord-GNN	Yes	WordNet	Yes	Yes	No
Proposed System (planned)	Yes	SBERT + WordNet	Yes	Yes	Planned

The comparison shows that graph-based methods add explicit relational structure, and that CL-UniDepWord-GNN is the only reviewed system that also uses syntax. None of the reviewed methods addresses AI-generated text, which is the gap the proposed system targets.

XIV. Strengths and Limitations

A. Strengths

- The ablation shows complementary value: the hybrid graph is best on three of four language pairs.
- A shared UD label set lets the same relation types describe sentences in different languages.
- Typed edges make the graph inspectable, even though the network's decisions are not.
- The reported training setup is small (64-dimensional embeddings, ten epochs), which suggests modest training cost on the corpora used.
- Explicit relations give a route to showing why two passages were matched, beyond a single score.
- The design extends naturally to further language pairs wherever a wordnet and a UD parser exist.

B. Limitations

- Quality depends on WordNet coverage and on parser accuracy for each language, and wordnet resources differ in size and licensing [13].
- Evaluation corpora are dominated by parallel texts, with negatives from different sources, which may overstate real-world performance.
- Baselines appear to be compared using their originally published figures rather than identical splits.
- No sensitivity analysis is given for the window size, the three-edge cap or the 63% threshold.
- Graph construction and training become costly on large corpora, which limits real-time use at institutional scale.
- The decisions of the GAT and MLP remain hard to interpret.
- Languages without a mature wordnet or treebank are not covered.

XV. Research Gap

Several problems remain open:

- Lexical systems have limited semantic understanding, and graph-based systems are not yet tested on realistic cases of translated plagiarism within full documents.
- Evaluation designs rarely separate genuine reuse from differences between source corpora.
- There is no mechanism for detecting AI-generated content in the reviewed systems.
- Reports offer little explanation or visualisation of why a passage was flagged.

- Graph-based models are computationally expensive.
- Support for low-resource languages is limited.
- Real-time deployment and adoption at institutional scale remain difficult.
- The predictions of graph neural networks are hard to interpret and explain.

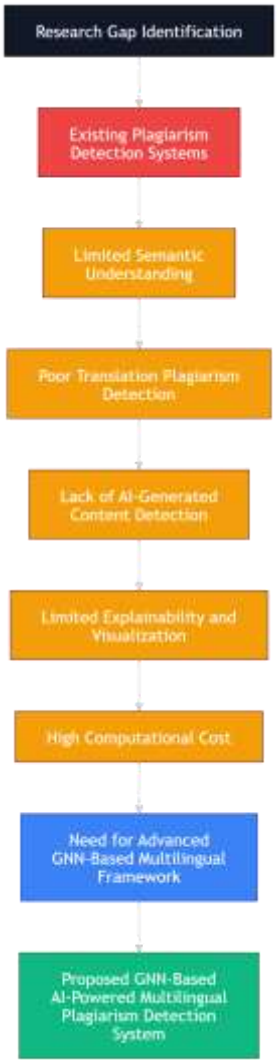


Fig. 6: Research Gap Identification

Fig. 6 traces the chain from the limitations of existing systems to the need for a GNN-based multilingual framework, which motivates the AI-powered system proposed next.

XVI. Proposed Future System

The proposed prototype combines a conventional web stack with the graph-learning components reviewed above, as shown in Fig. 7. A user submits documents through a React front end; a FastAPI back end handles authentication and storage in a MySQL database and calls the analysis modules. These are a Sentence-Transformer module that produces multilingual sentence embeddings [6], WordNet and Universal Dependencies analysis feeding a knowledge-graph builder [22], RotatE-style relation embeddings [8], GCN [9] and GAT [10] layers, a similarity analysis engine, an AI-generated-content detection module, a report generator and an interactive dashboard. The system aims to flag direct, semantic, translated and paraphrased reuse as well as machine-written text, and to explain its output through highlighted passages, knowledge-graph views and similarity heatmaps.

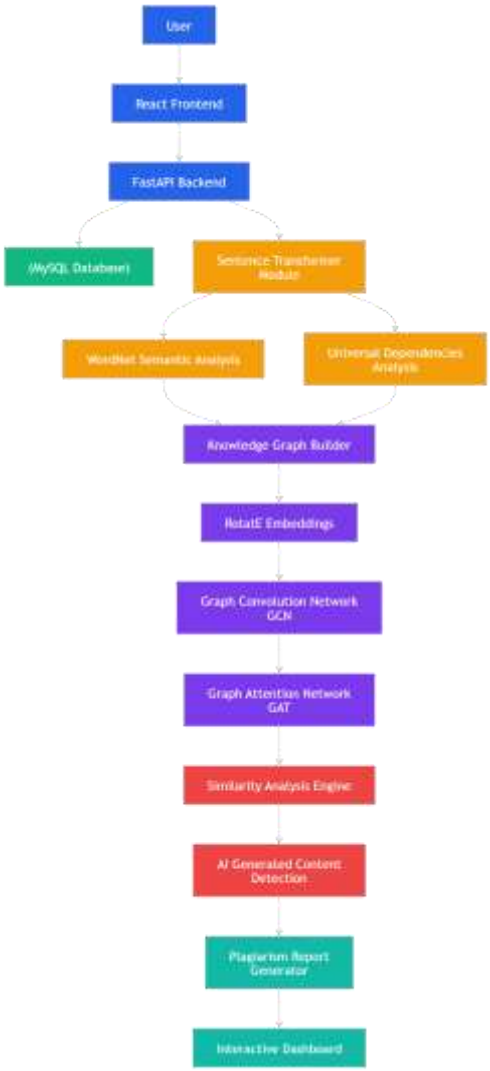


Fig. 7: Proposed AI-Powered Multilingual Plagiarism Detection System

Fig. 7 shows how these components connect.

Two cautions shape the design. Claims about accuracy should wait until the prototype has been tested on held-out sources with negatives drawn from the same collections as positives. And detection of machine-written text is an unsettled problem in which detectors can misclassify human writing, so that module should report a confidence level and should never be the only basis for a misconduct decision.

XVII. Future Research Directions

Explainable graph neural networks for plagiarism detection are the most pressing direction, since institutions need to justify a decision to a student. Scalability is the second: cloud deployment and cached graph construction could support near-real-time analysis of large submissions. Further topics include coverage of low-resource languages, detection strategies that use large language models, more reliable identification of AI-written text, and integration with learning-management platforms. Interpretable graph embeddings and transparent similarity scores would also help users trust and adopt these tools.

XVIII. Conclusion

The framework reviewed here shows that adding syntactic structure to WordNet-based graphs can improve cross-language plagiarism detection in three of four language pairs, with the language-dependent pattern in the ablation being its most informative result. The size of the reported gains over earlier systems should be treated with caution until they are confirmed on held-out corpora with like-for-like baselines. Semantic resources combined with graph learning remain a promising basis for next-generation detectors. Future systems should add reliable handling of AI-generated content, scalable deployment and explanation of their output, while preserving multilingual support.

References

[1] C. Bouaine, F. Benabbou, A. Bouaine, and C. Zaoui, "Multilingual Plagiarism Detection Using GNNs and Syntax-Semantic Knowledge Graphs," *IEEE Access*, vol. 14, pp. 99027–99043, 2026.
[2] O. Hourrane and E. H. Benlahmar, "Graph Transformer for Cross-Lingual Plagiarism Detection," *IAES Int. J. Artif. Intell. (IJ-AI)*, vol. 11, no. 3, pp. 905–915, Sep. 2022.
[3] M. Potthast, B. Stein, A. Eiselt, A. Barro´n-Ceden˜o, and P. Rosso, "PAN Plagiarism Corpus 2011 (PAN-PC-11)," Zenodo, 2011.
[4] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Proc. MT Summit X*, Phuket, Thailand, 2005, pp. 79–86.
[5] R. Steinberger et al., "The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages," in *Proc. 5th Int. Conf. Lang. Resour. Eval. (LREC)*, 2006.
[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.
[7] G. A. Miller, "WordNet: A Lexical Database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[8] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," in *Proc. ICLR*, 2019.
[9] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. ICLR*, 2017.
[10] P. Velickovic et al., "Graph Attention Networks," in *Proc. ICLR*, 2018.
[11] P. Qi et al., "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages," in *Proc. ACL System Demonstrations*, 2020.
[12] P. Vossen, "Introduction to EuroWordNet," *Comput. Humanities*, vol. 32, nos. 2–3, pp. 73–89, 1998.
[13] F. Bond and K. Paik, "A Survey of WordNets and Their Licenses," in *Proc. Global WordNet Conf.*, 2012.
[14] M. Franco-Salvador, P. Rosso, and M. Montes-y-Go´mez, "A Systematic Study of Knowledge Graph Analysis for Cross-Language Plagiarism Detection," *Inf. Process. Manage.*, vol. 52, no. 4, pp. 550–570, 2016.
[15] R. Navigli and S. P. Ponzetto, "BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network," *Artif. Intell.*, vol. 193, pp. 217–250, 2012.
[16] M. Potthast, B. Stein, A. Barro´n-Ceden˜o, and P. Rosso, "An Evaluation Framework for Plagiarism Detection," in *Proc. COLING*, 2010, pp. 997–1005.
[17] J. Nivre et al., "Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection," in *Proc. LREC*, 2020, pp. 4034–4043.
[18] J. Tiedemann, "Parallel Data, Tools and Interfaces in OPUS," in *Proc. LREC*, 2012, pp. 2214–2218.
[19] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity," in *Proc. AAAI*, 2006.
[20] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL*, 2019.
[21] A. Vaswani et al., "Attention Is All You Need," in *Proc. NeurIPS*, 2017.
[22] A. Hogan et al., "Knowledge Graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, 2022.
[23] T. Brown et al., "Language Models are Few-Shot Learners," in *Proc. NeurIPS*, 2020.
[24] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," in *Proc. ACL*, 2020.
[25] J. Zhou et al., "Graph Neural Networks: A Review of Methods and Applications," *AI Open*, vol. 1, pp. 57–81, 2020.
[26] J. Stegmu¨ller, F. Bauer-Marquart, N. Meuschke, T. Ruas, M. Schubotz, and B. Gipp, "Detecting Cross-Language Plagiarism Using Open Knowledge Graphs," Tech. Rep., 2021, doi: 10.6084/m9.figshare.17212340.v3.
[27] M. Franco-Salvador, P. Rosso, and R. Navigli, "A Knowledge-Based Representation for Cross-Language Document Retrieval and Categorization," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 414–423.