International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Developing an Intelligent Job Recommendation System Using Semantic Retrieval and Explainable AI Techniques

Hussein Ali Al Awad[1] Dr. Khaled Fathi Omar[1]

[1]Master of Web Science, Syrian Virtual University, Damascus, Syria

Abstract

The rapid growth of online recruitment platforms has created a need for job recommendation systems that can retrieve relevant opportunities from large and heterogeneous collections of job postings. Conventional keyword-based search remains efficient and transparent, but it often fails when equivalent job roles are expressed with different terms. This paper presents an intelligent job recommendation system that combines lexical retrieval, semantic retrieval, and explainable artificial intelligence techniques. The system is designed for a metadata-only setting and uses structured fields such as job title, company name, location, seniority level, job function, employment type, and industry. It does not rely on full job descriptions, user profiles, click logs, or application histories. The proposed pipeline builds a sparse lexical representation using TF-IDF and a dense semantic representation using Sentence-BERT embeddings. Candidate jobs are retrieved through semantic nearest-neighbor search and then ranked using a weighted hybrid scoring function. An optional Cross-Encoder re-ranking stage is used to refine the top candidates. To improve transparency, the system reports matched keywords, applied filters, and metadata-based evidence. Experiments on a cleaned LinkedIn job posting dataset containing 31,262 records show that the best hybrid configuration achieved Precision@10 of 0.8032 and nDCG@10 of 0.9496. A second-stage Cross-Encoder improved Precision@10 from 0.7896 to 0.7948 and nDCG@10 from 0.9666 to 0.9739 under the internal evaluation protocol. These findings indicate that a carefully engineered combination of lexical matching, semantic retrieval, and explainable ranking can produce effective job recommendations even when only structured metadata is available.

Keywords: Job recommendation system; semantic retrieval; explainable artificial intelligence; TF-IDF; Sentence-BERT; Cross-Encoder; hybrid ranking; information retrieval; natural language processing.

1 Introduction

Online recruitment platforms have become central channels for connecting job seekers with employers. These platforms contain large numbers of job opportunities that vary by title, company, location, seniority level, employment type, job function, and industry. As the number of postings grows, users often struggle to identify opportunities that match their goals and constraints.

Many recruitment search engines still rely heavily on keyword matching. This approach is computationally efficient and easy to interpret, yet it is limited when a query and a relevant job

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

posting use different surface forms. For example, a query for *software developer* may correspond to postings titled *software engineer*, *backend engineer*, or *application developer*. Purely lexical retrieval may under-rank these postings because it depends strongly on exact term overlap.

Recent progress in natural language processing has made semantic search practical through dense vector representations. Transformer-based models such as BERT and Sentence-BERT represent short texts in embedding spaces where semantic similarity can be measured using cosine similarity or inner product. These models improve retrieval by capturing meaning beyond exact keyword overlap. However, semantic similarity alone may overlook explicit job-search constraints such as location, employment type, seniority level, and work mode.

This paper proposes a hybrid and explainable job recommendation system that combines lexical and semantic retrieval. The system is intended for realistic metadata-only scenarios in which full job descriptions and user interaction histories are unavailable. The architecture supports preprocessing, document construction, embedding generation, semantic candidate retrieval, hybrid scoring, query-aware filtering, optional neural re-ranking, result diversification, and explanation generation.

1.1 Research Contributions

The main contributions of this work are as follows:

- A metadata-only job recommendation framework that does not require full job descriptions or historical user interactions.
- A hybrid retrieval model that combines TF-IDF lexical similarity and Sentence-BERT semantic embeddings.
- A semantic candidate generation strategy based on normalized dense vectors and nearest-neighbor retrieval.
- Query-aware filtering for employment type, seniority level, location, and remote, hybrid, or onsite work mode.
- An explainable AI layer that reports matched terms, applied filters, and metadata-based evidence.
- An empirical evaluation using Precision@10 and nDCG@10 on a cleaned LinkedIn job posting dataset.

2 Related Work

Recommendation systems are commonly grouped into collaborative filtering, content-based filtering, and hybrid methods. Collaborative filtering depends on interaction histories such as ratings, clicks, or applications. Although effective on mature platforms, it suffers from cold-start limitations when new users or new items have little historical data. Content-based recommendation uses item attributes and is more appropriate when interaction data is unavailable.

Job recommendation has often been treated as a content-based or hybrid retrieval problem. Structured metadata, including job title, industry, seniority, employment type, and location, provides useful signals for matching user intent with job postings. Traditional approaches such as Boolean retrieval and TF-IDF remain valuable because they preserve exact matching for important constraints such as *remote*, *senior*, *full-time*, or a specific city.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Transformer-based models have improved semantic similarity and information retrieval. BERT introduced bidirectional contextual representations that can be adapted to many NLP tasks [1]. Sentence-BERT extends this paradigm by producing efficient sentence-level embeddings suitable for semantic textual similarity and retrieval [2]. Vector indexes such as FAISS support efficient similarity search over dense embeddings at larger scale [3].

Neural re-ranking models provide an additional retrieval layer. Cross-Encoders jointly process the query and candidate document and return a direct relevance score. This can improve fine-grained ranking quality but is more computationally expensive than Bi-Encoder retrieval [4]. Therefore, Cross-Encoders are typically applied only to a limited set of top candidates.

Explainable AI has also become important in recommendation systems. In employment contexts, explanations matter because recommendations may influence career decisions. Common explanation strategies include matched keywords, score decomposition, feature-based evidence, rule-based filters, and concise natural-language rationales [5].

3 Problem Statement

The research problem addressed in this paper is the design of an accurate and explainable job recommendation system using only structured job metadata. The system must operate without full job descriptions, user profiles, click logs, application histories, or human-labeled relevance judgments.

This setting is challenging for several reasons. Job metadata is short, job titles are inconsistent across companies, abbreviations are common, and user queries may combine semantic intent with strict constraints. A query such as *remote junior data analyst London* contains a role, work mode, seniority level, and location. A useful system must understand the role semantically while also respecting explicit constraints.

The main technical challenges are:

- Keyword-only retrieval may miss semantically related jobs with different wording.
- Semantic retrieval may return broadly related jobs that violate exact user constraints.
- Metadata-only datasets provide less textual evidence than full job descriptions.
- Neural recommendation components require interpretable outputs to support user trust.
- Offline evaluation is difficult when human relevance labels are unavailable.

4 Proposed Methodology

The proposed methodology follows a pipeline-oriented design. First, raw job postings are validated and cleaned. Second, each job is converted into a composite text document built from the most informative metadata fields. Third, two indexes are created: a sparse lexical TF-IDF index and a dense semantic embedding index. Fourth, user queries are encoded using both representations. Fifth, semantic retrieval generates an initial candidate set. Sixth, lexical and semantic scores are normalized and combined into a hybrid score. Finally, filtering, diversification, optional Cross-Encoder re-ranking, and explanation generation are applied.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

4.1 Metadata Document Construction

Each job posting is represented by a composite metadata document:

$$d_i = [t_i, t_i, c_i, l_i, s_i, f_i, e_i, r_i], \tag{1}$$

where t_i is the job title, c_i is the company name, l_i is the location, s_i is the seniority level, f_i is the job function, e_i is the employment type, and r_i is the industry. The job title is repeated to increase its contribution because the title is usually the strongest signal in job search.

4.2 Lexical Similarity

The lexical component uses TF-IDF to represent metadata documents and the user query. The TF-IDF weight of term t in document d is defined as:

$$\text{tfidf}(t, d) = \text{tf}(t, d) \times \log\left(\frac{N}{\text{df}(t) + 1}\right), \tag{2}$$

where N is the number of documents and $\text{df}(t)$ is the number of documents containing term t. Lexical similarity is computed as a sparse dot product between the query vector and candidate document vectors.

4.3 Semantic Similarity

The semantic component uses the Sentence Transformer model `sentence-transformers/all-MiniLM-L6-v2`. Each document d_i and query q is encoded into normalized dense vectors $\mathbf{v}_{d_i}$ and $\mathbf{v}_q$. Semantic similarity is computed as:

$$s_{\text{sem}}(q, d_i) = \mathbf{v}_q^\top \mathbf{v}_{d_i}. \tag{3}$$

Because the embeddings are normalized, the inner product is equivalent to cosine similarity.

4.4 Hybrid Ranking

The lexical and semantic scores are min-max normalized before fusion:

$$\hat{s}(x) = \frac{s(x) - \min(s)}{\max(s) - \min(s) + \epsilon}. \tag{4}$$

The hybrid score is then computed as:

$$s_{\text{hybrid}}(q, d_i) = 0.4\, \hat{s}_{\text{sem}}(q, d_i) + 0.6\, \hat{s}_{\text{lex}}(q, d_i). \tag{5}$$

The lexical component receives a slightly larger weight because exact terms such as title keywords, locations, work mode, and seniority are important in job search. Semantic retrieval remains important because it expands the candidate set beyond exact keyword overlap.

4.5 Cross-Encoder Re-ranking

When re-ranking is enabled, the top candidates are processed as query-document pairs using `cross-encoder/ms-marco-MiniLM-L-6-v2`. The final score is computed as:

$$s_{\text{final}}(q, d_i) = \alpha\, \hat{s}_{\text{rerank}}(q, d_i) + (1 - \alpha)\, s_{\text{hybrid}}(q, d_i) + b_i, \tag{6}$$

where $\alpha = 0.7$ and b_i is an optional metadata bonus assigned when a candidate shares strong metadata evidence with the seed job in the evaluation setting.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

4.6 Implementation Details

The system was implemented in Python using `pandas`, `NumPy`, `scikit-learn`, `SciPy`, `joblib`, `sentence-transformers`, `PyTorch`, and `Streamlit`. Data preparation reads the LinkedIn Excel file, validates required columns, normalizes text, expands common abbreviations, removes noisy job titles, and saves the cleaned dataset as an artifact. Index training builds a TF-IDF matrix and Sentence-BERT embedding matrix over the composite metadata documents. Runtime recommendation loads the stored artifacts, retrieves semantic candidates using `NearestNeighbors` with cosine distance, computes lexical and semantic scores, applies hybrid fusion, optionally performs Cross-Encoder re-ranking, removes duplicates, limits repeated companies, and returns explainable recommendation fields. The Streamlit interface exposes the query, top-k value, filter toggle, re-ranking toggle, and company-diversity setting.

5 System Architecture

The architecture is organized as a layered retrieval and recommendation pipeline. Each layer performs a specific function and passes structured outputs to the next layer. This modular design supports maintainability, reproducibility, and controlled experimentation.

Table 1: Functional layers of the proposed job recommendation architecture.

Layer	Main Function	Output
Data ingestion	Reads and validates structured job metadata	Validated job records
Preprocessing	Normalizes text, expands abbreviations, and removes noisy titles	Cleaned metadata
Indexing	Builds TF-IDF vectors and Sentence-BERT embeddings	Sparse and dense indexes
Retrieval	Retrieves semantic candidates using vector similarity	Candidate job list
Hybrid ranking	Combines lexical, semantic, metadata, and optional re-ranking scores	Ranked recommendations
Explainability	Reports matched terms, filters, and metadata evidence	Transparent recommendation output

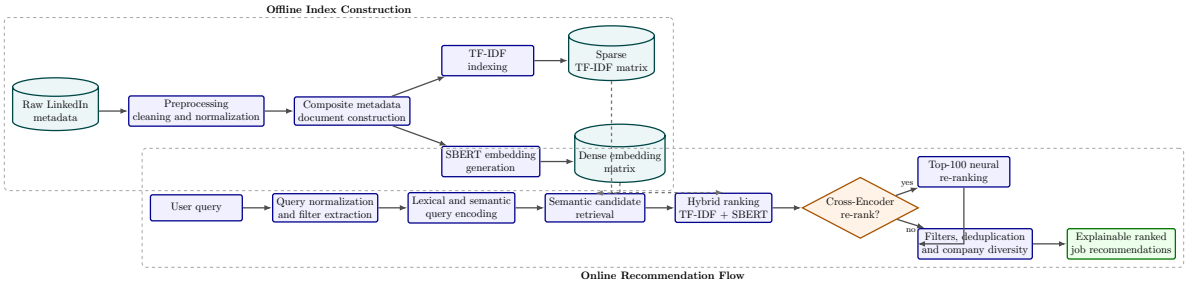

Figure 1: Overall architecture of the proposed intelligent job recommendation system.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

6 Retrieval Pipeline

The retrieval pipeline combines semantic candidate generation with hybrid ranking. Semantic retrieval is used first because it can identify related jobs even when exact lexical overlap is limited. Lexical evidence is then reintroduced during ranking to preserve exact matching for important query constraints.

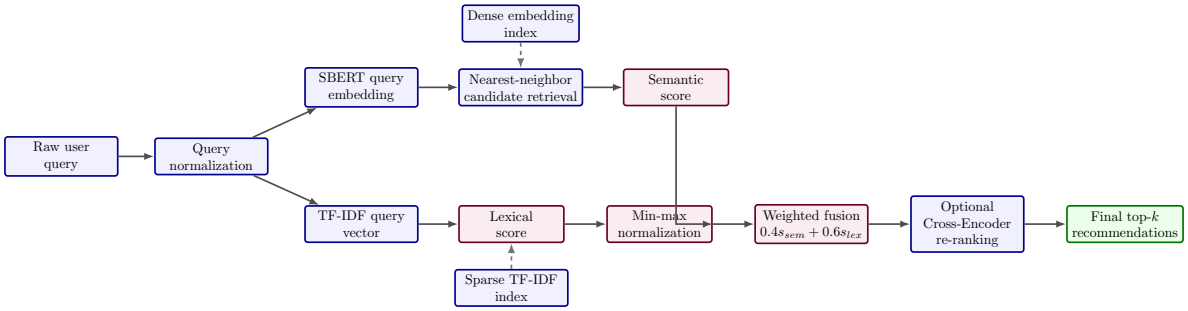

Figure 2: Hybrid retrieval pipeline combining semantic candidate generation and lexical-semantic ranking.

6.1 Query-Aware Filtering and Diversification

The system extracts explicit filters from the query when possible. Supported filters include employment type, seniority level, location hints, and work mode. If the extracted filters remove all candidates, the system falls back to the unfiltered candidate set to avoid returning an empty result list.

The final ranking stage also applies duplicate suppression using a tuple of normalized job title, company, and location. A company-level cap limits repeated results from the same employer. These rules improve result diversity without changing the core retrieval model.

7 Query Flow

The online query flow begins when the user submits a free-text job-search query. The query is normalized, encoded lexically and semantically, and used to retrieve an initial candidate set. Ranking and explanation are then applied before results are shown to the user.

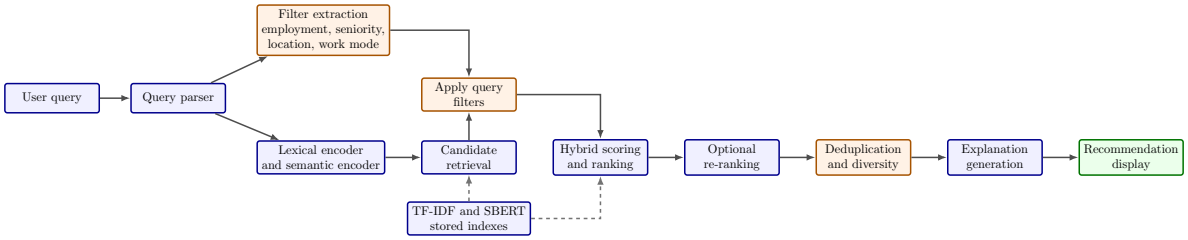

Figure 3: Online query flow from user input to explainable job recommendations.

8 Explainable AI Layer

The explainability component is intended to make recommendations understandable to end users. Instead of exposing only a numerical score, the system reports evidence that links the user query to the recommended job. This is particularly important in job recommendation because

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

users often need to know whether a result matches their role, seniority, location, employment type, or work-mode constraints.

The explanation layer includes:

- **Keyword overlap:** shared terms between the user query and the composite job document.
- **Filter explanation:** extracted constraints such as remote, hybrid, junior, full-time, or a location term.
- **Metadata evidence:** matches in job function, industry, seniority level, or employment type.
- **Ranking evidence:** indication of whether a result was mainly supported by lexical matching, semantic similarity, or neural re-ranking.

For example, for the query *remote junior data analyst London*, a recommended job may include the explanation: matched keywords are *data*, *analyst*, *remote*, and *London*; applied filters are remote work mode, junior seniority, and location. This does not fully interpret the internal embedding space, but it provides practical and readable transparency.

9 Dataset and Preprocessing

The evaluation used a LinkedIn job posting dataset. The raw dataset contained 31,597 records. After cleaning and validation, 31,262 valid records remained. The system used only structured metadata fields: job title, company name, location, hiring status, date, seniority level, job function, employment type, and industry.

The preprocessing pipeline included missing-value handling, whitespace normalization, job-title normalization, abbreviation expansion, and removal of noisy records. Examples of abbreviation expansion include *ML* to *machine learning*, *AI* to *artificial intelligence*, *SWE* and *SDE* to *software engineer*, *QA* to *quality assurance*, *PM* to *product manager*, and *MLE* to *machine learning engineer*.

Table 2: Dataset fields used by the metadata-only recommendation system.

Field	Role in the system
Job title	Primary role signal and strongest lexical evidence
Company name	Employer metadata and duplicate-control feature
Location	Geographic and work-location matching
Hiring status	Structured posting status metadata
Date	Temporal metadata retained from the source dataset
Seniority level	Career-level filtering and relevance grading
Job function	Functional category for matching and evaluation
Employment type	Full-time, part-time, contract, internship, and related filters
Industry	Domain-level evidence for matching and relevance grading

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Table 3: Dataset size before and after preprocessing.

Stage	Number of Records	Description
Raw dataset	31,597	Original LinkedIn job postings
Cleaned dataset	31,262	Valid records after preprocessing and noise removal
Removed records	335	Invalid, noisy, or unusable records

10 Experimental Evaluation

The evaluation was conducted using an internal metadata-based relevance protocol because human relevance judgments and user interaction logs were unavailable. Seed jobs were sampled from the dataset, and their normalized job titles were used as queries. Retrieved jobs were compared with the seed job using metadata consistency.

Relevance labels were assigned as follows: 3 for the same normalized job title, 2 for the same job function or industry, 1 for the same seniority level or employment type, and 0 otherwise. Precision@10 considered labels 2 and 3 as relevant. nDCG@10 was used to evaluate ranking quality with graded relevance.

Precision@10 is defined as:

$$\text{Precision@10} = \frac{|\{d_i \in R_{10} : rel(d_i) \geq 2\}|}{10}, \tag{7}$$

where R_{10} is the set of the top 10 returned jobs.

Discounted cumulative gain at rank k is computed as:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}. \tag{8}$$

The normalized version is:

$$\text{nDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}, \tag{9}$$

where IDCG@k is the ideal DCG obtained by sorting results by true graded relevance.

Table 4: Internal relevance grading protocol used for offline evaluation.

Grade	Condition	Interpretation
3	Same normalized job title	Highly relevant
2	Same job function or same industry	Relevant
1	Same seniority level or same employment type	Weakly related
0	None of the above metadata matches	Not relevant

10.1 Hybrid Retrieval Results

Table 5 reports the hybrid retrieval results across different candidate sizes and weighting settings. The best reported setting used 250 semantic candidates with semantic weight 0.4 and lexical weight 0.6, achieving Precision@10 of 0.8032 and nDCG@10 of 0.9496.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Table 5: Hybrid retrieval performance under different candidate sizes and weighting configurations.

Candidate Size	Semantic Weight	Lexical Weight	Precision@10	nDCG@10
80	0.7	0.3	0.7624	0.9360
80	0.6	0.4	0.7676	0.9388
80	0.5	0.5	0.7780	0.9392
80	0.4	0.6	0.7884	0.9432
150	0.7	0.3	0.7640	0.9365
150	0.6	0.4	0.7776	0.9413
150	0.5	0.5	0.7844	0.9420
150	0.4	0.6	0.7984	0.9478
250	0.7	0.3	0.7680	0.9393
250	0.6	0.4	0.7832	0.9389
250	0.5	0.5	0.7936	0.9454
250	0.4	0.6	**0.8032**	**0.9496**

10.2 Cross-Encoder Re-ranking Results

Table 6 compares the baseline hybrid ranking with the optional Cross-Encoder re-ranking stage. Re-ranking was applied to the top 100 candidates using $\alpha = 0.7$. The Cross-Encoder produced modest but consistent improvements in both metrics.

Table 6: Baseline hybrid ranking versus Cross-Encoder re-ranking.

Configuration	Precision@10	nDCG@10	Notes
Baseline hybrid ranking	0.7896 ± 0.2896	0.9666 ± 0.1051	No Cross-Encoder
Cross-Encoder re-ranking	0.7948 ± 0.2946	0.9739 ± 0.1046	Top 100 candidates
Delta	+0.0052	+0.0072	Rerank minus baseline

11 Discussion

The results show that a hybrid retrieval design is appropriate for metadata-only job recommendation. Larger semantic candidate sets improved the likelihood that relevant jobs were available to the ranking stage. At the same time, configurations with stronger lexical weighting performed best, which is consistent with the short and structured nature of job metadata. Exact words in job titles, locations, seniority levels, and employment types remain highly informative.

The Cross-Encoder re-ranking stage improved both Precision@10 and nDCG@10, although the gains were moderate. This is expected because the base hybrid model already performs strongly under the metadata-derived relevance protocol. The benefit of re-ranking is that it provides a more detailed interaction between the query and candidate document, while its drawback is increased computational cost.

The explanation layer provides practical transparency by showing matched keywords and applied filters. This form of explanation is intentionally simple. It does not claim to fully explain dense embedding behavior, but it gives users evidence that is directly connected to the visible

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

metadata.

11.1 Limitations

This study has several limitations. First, the system uses structured metadata only and does not include full job descriptions, skill requirements, salary, education level, or company descriptions. Second, the evaluation uses heuristic relevance labels derived from metadata consistency rather than human judgments or real user interactions. Third, the models are pre-trained general-purpose models and were not fine-tuned on a job-specific relevance dataset. Fourth, Cross-Encoder re-ranking improves quality but increases latency and computational cost. Finally, the dataset represents a fixed snapshot of job postings, while real recruitment platforms require continuous updates as jobs are posted, modified, and closed.

12 Conclusion

This paper presented an intelligent job recommendation system that combines semantic retrieval, lexical matching, explainable AI techniques, and optional neural re-ranking. The system was designed for a practical metadata-only setting where full descriptions and user interaction histories are unavailable. The proposed pipeline uses TF-IDF for exact lexical evidence, Sentence-BERT for semantic candidate retrieval, weighted hybrid scoring for ranking, query-aware filters for explicit constraints, and simple explanation fields for transparency.

Experiments on a cleaned LinkedIn job dataset showed that the best hybrid configuration achieved Precision@10 of 0.8032 and nDCG@10 of 0.9496. Cross-Encoder re-ranking further improved Precision@10 from 0.7896 to 0.7948 and nDCG@10 from 0.9666 to 0.9739. The results support the conclusion that effective and interpretable job recommendation is possible even under limited data conditions, provided that preprocessing, retrieval, ranking, filtering, and evaluation are carefully engineered.

Future work should include human relevance judgments, real user interaction data, full job descriptions, skill extraction, learning-to-rank models, fairness analysis, and deployment-oriented latency evaluation.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of EMNLP-IJCNLP*, 2019, pp. 3982–3992.

[3] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[4] N. Thakur, N. Reimers, A. Ruckle, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Proceedings of NeurIPS Datasets and Benchmarks*, 2021.

[5] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[6] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.

[7] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[8] A. Vaswani et al., "Attention is all you need," in *Proceedings of NeurIPS*, 2017, pp. 5998–6008.

[9] J. Lin, R. Nogueira, and A. Yates, "Pretrained transformers for text ranking: BERT and beyond," *Synthesis Lectures on Human Language Technologies*, vol. 14, no. 4, pp. 1–325, 2021.

[10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of ICML*, 2005, pp. 89–96.