An Interpretable AI Framework for Resume Ranking via Hybrid NLP [3] and Large Language Models

Mahua Mukhopadhyay
University of Southampton

***Abstract*—The increasing volume of job applications has made manual resume screening inefficient, time-consuming, and prone to inconsistency. Existing automated solutions largely rely on either keywork-based matching, which lacks contextual understanding, or large language model (LLM)-based approaches, which often suffer from limited interpretability and potential bias. To address these limitations, this paper proposes a hybrid framework that integrates traditional Natural Language Processing (NLP [3]) techniques with LLM- driven semantic analysis for automated resume ranking.**

The proposed system processes curriculum vitas and job descriptions across multiple document formats, extracting structured features such as skills, experience, and qualifications using NLP [3] tools while simultaneously leveraging LLMs to capture contextual and semantic relationships. A hybrid similarity scoring mechanism is introduced, combining keyword-based matching with embedding-based semantic similarity to generate robust candidate rankings. This approach enables improved alignment between candidate profiles and job requirements while maintaining interpretability in the ranking process

Experimental evaluation demonstrates that the proposed method achieves improved matching performance compared to traditional keyword-based systems and offers greater transparency than purely LLM-based approaches. The results highlight the effectiveness of combining symbolic and neural techniques in development of scalable, interpretable, and efficient AI-driven hiring system.[15]

I. INTRODUCTION

The rapid digitization of recruitment processes has led to a significant increase in the volume of job applications submitted to organizations across industries. As result, recruiters are often required to manually review hundreds or even thousands of curriculum vitae (CVs) for a single position. This process is not only time-consuming but also prone to human error, inconsistency, there is a growing demand for intelligent automated systems capable of assisting or replacing manual resume screening.

Early approaches to automated resume evaluation primarily relied on keyword-based filtering and information retrieval techniques. Methods such as term frequency-inverse document frequency (TF-IDF [5]), introduced by Gerard Salton, and statistical term weighting frameworks proposed by Karen Spärk Jones have been widely used to identity relevant terms within documents[11]. While these techniques offer computational efficiency and simplicity, they lack the ability to capture semantic relationships between words and phrases. As a result, they often fail to recognize equivalence between related concepts expressed using different terminology, leading to suboptimal candidate matching.

The emergence of deep learning and advances in Natural Language Processing have significantly transformed text analysis and information extraction tasks[10]. In particular, the introduction of the Transformer architecture by Ashish Vaswani et al. [2] marked a breakthrough in sequence modelling by enabling contextual understanding of entire documents. Building on this architecture, models such as BERT developed by Jocob Devlin et al. [1] have demonstrated state-of-the art performance in a wide range of NLP [3] tasks, including sematic similarity, named entity recognition, and document classification. These models allow systems to move beyond surface-level keyword matching and instead capture deeper linguistic and contextual relationships.

More recently, large language models (LLMs), such as those developed by OpenAI [8], have further advanced the capabilities of automated text understanding. These models can perform complex reasoning, summarization, and semantic comparison tasks with

minimal task- specific training. In the context of recruitment, LLMs have been applied to tasks such as resume parsing, candidate evaluation, and job matching[12]. However, despite their strong performance, LLM-based systems present several practical challenges. These include high computational cost, limited interpretability, and concerns regarding fairness and bias in decision-making processes.[14][15]

Given these limitations, there is a clear need for approaches that balance the strength of traditional NLP [3] techniques with the advanced capabilities of modern LLMs. Specifically, system should aim to combine the interpretability and efficiency of rule-based methods with the semantic understanding provided by deep learning models. Such a hybrid approach has the potential to improve matching accuracy while maintaining transparency and scalability.

In this paper, we propose a hybrid framework for automated resume ranking that integrates structured feature extraction with LLM-based semantic analysis. The system processes CVs and job descriptions in multiple formats, extracts relevant features such as skills and experience. and computes similarity scores using a combination of keyword-based and embedding-based methods. By leveraging both symbolic and neural representations, the proposed approach aims to provide more accurate and interpretable candidate rankings.

The main contributions of the work are as follows:

1. A multi-format document processing pipeline for CV and job description analysis
2. A hybrid feature extraction approach combining traditional NLP [3] technique with LLM-based semantic approach
3. A weighted similarity scoring mechanism for robust candidate ranking
4. A scalable and interpretable framework suitable for real- world recruitment applications.

The remainder of this paper is organized as follows:
Section 2 reviews related work in automated resume screening and semantic matching, Section 3 describes the proposed methodology, section 4 presents experimental evaluation and results, and Section 5 concludes the paper with future research directions.

II RELATED WORK

2.1. Traditional Resume Screening and Information Retrieval Approaches

Early research in automated resume screening is rooted in classical information retrieval techniques, where documents are represented as vectors of term frequencies. Methods such as TF-IDF, introduced by Gerard Salton [5], and probabilities term weighting models developed by Karen Spärk Jones[11] formed the foundation for keyword-based matching systems. These approaches compute similarity between documents using measures such as cosine similarity, enabling basic ranking of candidates based on overlapping terms.[6]

While effective in structured and domain-specific environments, these methods exhibit significant limitations when applied to real-world recruitment scenarios. They fail to capture semantic relationship between terms, making them sensitive to variations in words and phrasing. For example, similar skills may be incorrectly ranked lower. Additionally manual tuning of rules and dictionaries reducing scalability and adaptability.

2.2 Word Embeddings and Early Neural Approaches

To address the limitation of keyword-based methods, researchers introduced distributed word representations, such as Word2Vec proposed by Tomas Mikolov et at. And GloVe [4] developed by Jeffery Pennington et al. These models represent words as dense vectors in a continuous space, capturing semantic similarity based on contextual usage. As a result, they enable systems to identify relationships between words that do not explicitly match.

Subsequent work extended these representations to sentence and document-level embeddings, allowing for more effective comparison of longer text segments. However, these approaches still rely on relatively shallow contextual representations and documents. In the context of resume screening [12], this limits their ability to fully capture the complexity of candidate profiles and job requirements.

2.3Transformer-Based Models and Contextual Understnading

The introduction of the Transformer architecture by Ashish Vaswani et al. marked a significant

advancement in natural language processing by enabling models to capture contextual relationships scores entire sequences. Unlike earlier models, transformers [2] utilize self-attention mechanisms to weigh the importance of different words within a document, resulting in richer and more dynamics representations.

Building of this architecture, models such as BERT developed by Jacob Devlin et al [1]. have achieved state-of-art performance across a wide range of NLP [3] tasks. In recruitment applications transformer-based models have been used for resume classification, skill extraction[10], and semantic similarity computation. Furthermore, approaches such as Sentence-BERT [7], proposed by Nils Reimers and Iryna Gurevych, enable efficient computation of particularly suitable for comparing CVs and job descriptions.

Despite these advancements, transformer-based systems often require significant computational resources and large amounts of training data. Additionally, while they improve semantic understanding, they do not inherently address issues related to interpretability or transparency in decision-making.

2.4 Large Language Models in Recruitment Systems

Recent developments in large language models (LLMs), such as those developed by OpenAI [8], have further expanded the capabilities of automated text analysis system. These models can perform complex reasoning tasks, including summarization, interference, and contextual comparison, making them highly suitable for recruitment applications.

Several recent studies have explored the use of LLMs for end-to-end resume screening, where models are used ti directly revaluate and rank candidates based on job descriptions. These systems often leverage prompt-based or multi-agent architectures, allowing different components to handle tasks such as information extraction, scoring, and explanation generation. While these approaches demonstrate strong performance, they also introduce

2.5 Hybrid Approaches and Research Gap

Given the limitations of both traditional NLP [3] methods and LLM-based systems, recent research has begun to explore hybrid approaches that combine symbolic and neural techniques. These approaches aim to leverage the strengths of each paradigm: the interpretability and efficiency of rule-based methods, and the contextual under of deep learning models.

However, existing hybrid systems remain relatively underexplored, particularly in the context of resume ranking. Many proposed solutions either rely heavily on LLMs with minimal interpretability or incorporate basic keyword matching without effectively integrating without efficiently integrating semantic representations [15]. As a result, there is still a lack of frameworks that provide a balanced combination of accuracy, transparency, and scalability.

This paper addressed this gap by proposing a hybrid NLP [3]-LLM framework that integrates structured feature extraction with semantic similarity modelling.[15] Unlike prior approaches, the proposed system explicitly combines keyword-based and embedding-based similarity measures through a weighted scoring mechanism, enabling both precise and context-aware candidate ranking. Furthermore, the modular design of the system enhances interpretability and adaptability, making it suitable for real-world deployment in recruitment pipelines.

III METHODOLOGY

The proposed system is designed as hybrid pipeline that integrates traditional Natural Language Processing (NLP [3]) techniques with large language model (LLM)-based semantic analysis to perform automated resume ranking. The methodology focuses on transforming unstructured textual data into structured representations, computing similarity scores between candidate profiles and job descriptions, and producing an interpretable ranking of candidates.

The overall pipeline consists of four primary stages:

1. Document parsing and preprocessing
2. Feature extraction
3. Semantic representation, and
4. Similarity computation and candidate ranking.

3.1 System Overview

The system operates as a multi-stage processing pipeline that takes as input a collection of CVs and a target job description. Each document is processed through parallel pipelines that extract both structured features and semantic representations. These outputs are then combined in a hybrid similarity framework to

compute in a hybrid similarity framework to compute a final ranking score.

The design emphasizes:

- Modularity, allowing independent optimization of components.
- Interpretability, through explicit feature matching
- Scalability, for handling large candidate

3.2 Document Parsing and Preprocessing

The first stage involves extracting textual content from CVs and job descriptions in the multiple formats, including PDF, DOC, and DOCX. Specialized parsing tools are used to ensure accurate extraction while preserving structural elements such as headings and lists.

Once extracted, the text undergoes preprocessing to improve consistency and reduce noise. This includes:

- Tokenization of text into words or sub words
- Conversion to lowercase for uniformity
- Removal of words and punctuation
- Normalization of text

These steps ensure that downstream processing is not affected by inconsistencies in formatting or writing style, which are common in real-world CVs.

3.3 Feature Extraction

Following preprocessing, the system performs structured feature extraction to identify key elements within each document. This is achieved using rule-based NLP [3] techniques implemented spaCy [9].

The extracted features include:

- Technical and soft skills
- Job titles and roles
- Educational qualifications domain-specific keywords

These features are represented as structured sets, enabling direct comparison between candidate CVs and job requirements. This stage is particularly important for interpretability, as it allows the system to explicitly identify which requirements are satisfied by each candidate.

3.4 Semantic Representation

While structured features provide clarity, they do not capture deeper contextual relationships between different expressions of similar concepts. To address this limitation, the system incorporates semantic representation using transformer-based models such as BERT [1].

Each document is transformed into a dense vector representation (embedding) that encodes contextual meaning. These embeddings allow the system to:

- Recognize synonyms and related concepts
- Capture relationships between phrases
- Compare documents beyond surface-level keyword matching

This semantic layer significantly improves the system's ability to match candidates whose skills are described using varied terminology.

3.5 Hybrid Similarity Framework

The core of the proposed methodology lies in combining structured feature matching with semantic into a unified framework. Rather than relying solely on integrates both approaches to achieve a balance precision and contextual understanding.

The hybrid similarity framework:

- Uses feature overlap to ensure explicit requirement matching
- Leverages semantic embeddings to capture contextual similarity
- Combines both signals into a single relevance score.

This design allows the system to adapt to different recruitment scenarios. For example, roles requiring strict technical skills can emphasize feature matching, while roles requiring broader competencies can benefit from semantic analysis.

3.6 Candidate Ranking and Output Generation

Once similarity scores are computed, candidates are ranked in descending order based on their relevance to the job description. The raking process ensures that the most suitable candidates appear at the top of the list.[13]

In addition to ranking, the system generates interpretable outputs by:

- Highlighting matched skills and qualifications
- Identifying missing requirements
- Providing similarity scores for transparency

These outputs can be visualized or exported enabling recruiters to make informed decisions efficiently.

3.7 System Design Considerations

The proposed framework is designed with several practical considerations in mind:

Scalability
The modular architecture allows the system to handle large volumes of CVs by parallelizing processing steps such as embedding generation and similarity computation.
Interpretability
By combining structured feature extraction with semantic analysis, the system provides both quantitative scores and qualitative explanations, addressing the transparency limitations of purely LLM-based approaches.
Flexibility

The hybrid design enables easy adjustment of system parameters, such as weighting between keyword and sematic similarity, making it adaptable to different job roles and industries.

4 MATHEMATICAL MODEL

This section formalizes the proposed hybrid framework as a similarity-based ranking model that integrates both symbolic feature representations and continuous semantic embeddings. The objective of the model is to quantity the relevance between a candidate's CV and given job description by combining explicit feature overlap with contextual semantic similarity.

4.1 Document Representation

Let the dataset consist of a collection of candidate CVs:

$$C = \{C_1, C_2, \dots \dots, C_N\}$$

Each document, weather a CV or job description, is initially represented in raw textual form. To ensure consistency and reduce noise, a preprocessing function P is applied to transform each document into a normalized textual representation:

$$T_i = P(D_i)$$

This preprocessing stage includes tokenization, and removal of irrelevant tokens. The resulting representation T_i serves as the foundation for both feature extraction and sematic embedding.

4.2 Feature-Based Representation

To capture interpretable and structured information, each document is mapped to a set of extracted features. These features include domain-relevant entities such as technical skills, job roles, and educational qualifications. Formally, each document is represented as:

$$F_i = \{f_{i1}, f_{i2}, \dots \dots, f_{ik}\}$$

Where each f_{ij} corresponds to a distinct feature identified through NLP [3]-based extraction.
The similarity between a candidate CV C and a job description J is first evaluated using keyword-based metric that measures the degree of overlap between their respective feature sets. This is defined as:

$$KeywordSim(C,J) = \frac{|F_C \cap F_J|}{|F_J|}$$

This formulation reflects the proportion of required job features that are satisfied by the candidate. By normalizing with respect to the job description, the metric emphasizes how well a candidate meets the specified requirement rather than simply counting raw matches.

4.3 Semantic Representation

While feature-based representations provide interpretability, they are limited in their ability to capture sematic relationships between different expressions of similar concepts. To address this limitation, each document is additionally madded into a continuous vector space using an embedding function ε_i typically implemented transformed transformer-based models such as BERT [1].

$$V_I = \varepsilon(T_i)$$

These embeddings encode contextual information, enabling the system to recognize semantic equivalence between phrases that may not share explicit keywords. For instance, terms such as "data analyst" and "business intelligence specialist" may be mapped to nearby regions in the embedding space.
The semantic similarity between a candidate CV and job description is computed using cosine similarity:

$$SemanticSim(C,J) = \frac{V_C \,.\, C_J}{||V_C||\,||V_J||}$$

This metric measures the angular similarity between the two vectors, providing a normalized indication of their contextual alignment.

4.4 Hybrid Scoring Function

To effectively combine the strengths of both representations, the proposed model introduces a

hybrid scoring function that integrates keyword-based and semantic measures into a unified framework.

This weighted combination provides flexibility, allowing the system to be adapted to different recruitment scenarios. For example, technical roles with strict skill requirement may prioritize keyword similarity, whereas roles requiring broader conceptual understanding may benefit from higher semantic weighting.

4.5 Candidate Ranking Function

Given the hybrid similarity score, each candidate C_i is assigned a relevance score with respect to the job description:

$$S_i = Score(C_I, J)$$

Candidates are then ranked in descending order based on their scores:

$$Rank = argsort(S_i)$$

This ranking function produces an ordered list of candidates, where higher-ranked individuals exhibit stronger alignment with job requirements.

4.6 Score Normalization

To ensure comparability across different job descriptions and candidate pools, the raw scores are normalized using min-max scaling:

$$S_i' = \frac{S_I - \min(S)}{\max(S) - \min(S)}$$

This normalization step constraints scores to a fixed range, improving interpretability and enabling consistent thresholding for candidate selection.

4.7 Interpretability and Explanation Function

A distinguishing feature of the proposed model is its emphasis on interpretability. In addition to generating ranking scores, the system provides explanations by identifying overlapping features between a CV and a job description.

Formally, an explanation function is defined as:

$$\boldsymbol{X(C,J) = (F_C \cap F_J, Score(C,J))}$$

This function outputs both the set of matched features and the corresponding similarity score, enabling users to understand the rationale behind each ranking decision. This transparency is particularly important in recruitment applications, where accountability and fairness are critical considerations.

V EXPERIMENTAL RESULTS AND EVALUATION

5.1 Experimental Setup

To evaluate the effectiveness of the proposed hybrid, resume ranking framework, experiments were conducted on a curated dataset consisting of CVs and job descriptions collected from publicly available templates and anonymized samples. The dataset includes 75 CVs across multiple domains, including software engineering, data science, and web development, along with 5 representative job descriptions.

Each CV was manually reviewed to establish a ground truth ranking, where candidates were labelled based on their relevance to each job description. This ground truth serves as a reference for evaluating the performance of the proposed system.

The system was implemented in Python using standard NLP [3] and data preprocessing libraries, including spaCy for feature extraction and transformer-based models such as BERT [1] for semantic embedding.

5.2 Evaluation Metrics

To assess ranking performance, the following evaluation metrics were used:

- Precision: Measures the proportion of relevant candidates in the top K results
- Recall: Measures how many relevant candidates are retrieved in the top K
- Normalized Discounted Cumulative Gain: Evaluates ranking quality by considering the position of relevant candidates
- Mean Reciprocal Rank: Measures how early the first relevant candidate appears.

These metrics are widely used in information retrieval and ranking systems to evaluate both accuracy and ranking quality.

5.3 Baseline Methods

To validate the effectiveness of the proposed approach, we compare it against two baseline methods:

1) Keyword-based Model (TF-IDF [5])

Traditional approach using term matching and cosine similarity

2) LLM-Only Semantic Model

Ranking based solely on embedding similarity without structured feature extraction

3) Proposed Hybrid Model

Combination off keyword-based and semantic similarity.

5.4 Quantitative Results

Table 1: Perform Comparison Across Models

Model	Percision@5	Recall@5	nDCG@5	MRR
Keyword-Based (TF-IDF [5])	0.62	0.55	0.6	0.58
LLM-Only Model	0.78	0.72	0.75	0.73
Proposed Hybrid Model	0.86	0.81	0.84	0.82

The result demonstrate that the proposed hybrid model consistently outperforms both baseline approaches across all evaluation metrics.

- Compared to the keyword-based model, the hybrid approach improves Precision@5 by approximately 24%, indicating significantly better candidate selection accuracy.
- Compared to the LLM-Only model, the hybrid inclusion of structured feature matching, which improves precision in cases where explicit skill requirements are critical.
- The improvement in nDCG indicates that relevant that relevant candidates are ranked higher in the list, which is crucial in real-world recruitment scenarios.

5.5 Ablation Study

To further understand the contribution of each component, an ablation study was conducted.

Table 2: Component Contribution Analysis

Configuration	Pecision@5	nDCG@5
Keyword Only	0.62	0.60
Semantic Only	0.78	0.75
Hybrid ($\alpha = 0.5, \beta = 0.5$)	0.83	0.81
Hybrid ($\alpha = 0.4, \beta = 0.5$)	0.86	0.84

Insights

- The hybrid model consistently outperforms individual components.
- Increasing the weight of semantic similarity slightly improves performance.
- However, removing keyword matching reduces interpretability and person

This confirms that both components contribute meaningfully to the overall system.

5.6 Qualitative Evaluation

In addition to quantitative analysis was conducted to evaluate the interpretability of the system.

The hybrid model provides:

- Clear identification of matched skills
- Explanation of ranking decisions
- Visibility into missing qualifications

For example, candidates ranked higher by the system consistently demonstrated stronger alignment with both required skills and contextual job requirements, while lower-ranked candidates lacked either explicit skills or semantic relevance.

5.7 Performance and Efficiency

The system demonstrates efficient processing performance

- Average processing time per CV: 0.8-1.2 seconds
- Batch processing significantly reduces total runtime
- Embedding computation is the most resource-intensive step

Despite the inclusion of LLM-based components, the hybrid system remains scalable due to modular architecture.

5.8 Discussion of Results:

- The Keyword-based systems alone are insufficient for capturing semantic relationships
- LLM-Only systems, while powerful, may overlook explicit requirements
- The hybrid model successfully integrates both approaches, resulting in improved performance.

These finding highlight the importance of combining symbolic and neural methods in practical AI systems,

particularly in domains such as recruitment where both accuracy and interpretability are essential.

VI. CONCLUSION AND FUTURE WORK

This paper presented a hybrid framework for automated resume ranking that integrates traditional Natural Language Processing (NLP [3]) techniques with large language model (LLM)-model semantic analysis. The proposed system addresses key limitations of existing approaches by combining structured feature extraction with contextual embeddings, both precise and semantically rich candidate-job matching.

The methodology introduced a modular pipeline capable of processing multi- format CVs and job descriptions, extracting interpretable features, and computing similarity scores through a hybrid scoring mechanism. By leveraging both keyword-based matching and performance. The mathematical formulation of the model further ensures clarity, reproductivity, and adaptability across different recruitment scenarios.

Experimental evaluation demonstrated that the proposed hybrid approach outperforms traditional keyword-based methods are purely LLM-based systems across multiple ranking metrics, including Precison@K, nDCG, and Mean Reciprocal Rank. The results indicates that candidate rankings. Additionally, the system provides interpretable outputs, enabling users to understand the rationale behind ranking decisions – an important requirement in real-world hiring processes.

Despite these contributions, several limitations remain. The performance of the system is influenced by the quality and diversity of input CVs, which can vary significantly in real-world setting. Furthermore, while the use of LLMs enhances semantic understanding, it may introduce biases inherited from training data. Computational cost associated with embedding generation is another factor that may affect scalability in large-scale deployments.

Future work will focus on addressing these challenges by incorporating bias mitigation techniques, optimizing model efficiency, and expanding the dataset to include more diverse and real-world examples. Additional enhancements may include integrating domain-specific fine-tuned language models, incorporating feedback loops from recruiters, and extending the framework to support related tasks such as candidate shortlisting, interview recommendation, and skill gap analysis. JDe19

Overall, this work demonstrates that hybrid NLP [3]-LLM approaches offer a promising direction for developing scalable, interpretable, and effective AI-driven recruitment systems, contributing to the broader advancement of intelligent decision-support tools in human resource management.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT [1]: Pre-training of Deep Bidirectional Transformers [2] for Language Understanding,” Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), pp. 4171–4186, 2019.

[2] Ashish Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[3] Thomas Wolf et al., “Transformers [2]: State-of-the-Art Natural Language Processing,” Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, 2020.

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, “GloVe [4]: Global Vectors for Word Representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP [3]), pp. 1532–1543, 2014.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean, “Efficient Estimation of Word Representations in Vector Space,” International Conference on Learning Representations (ICLR), 2013.

[6] Quoc V. Le and Tomas Mikolov, “Distributed Representations of Sentences and Documents,” Proceedings of the 31st International Conference on Machine Learning (ICML), pp. 1188–1196, 2014.

[7] Nils Reimers and Iryna Gurevych, “Sentence-BERT [1]: Sentence Embeddings using Siamese

BERT [1]-Networks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP [3]), pp. 3982–3992, 2019.
[8] OpenAI, "GPT-4 Technical Report," 2023.
[9] spaCy, "Industrial-Strength Natural Language Processing in Python," Explosion AI, 2023. [Online]. Available: https://spacy.io
[10] Gerard Salton and Christopher Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.
[11] Karen Spärck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11–21, 1972.
[12] "Automated Resume Screening Using Natural Language Processing and Machine Learning," Expert Systems with Applications, vol. 146, 2020.
[13] "Deep Learning Approaches for Resume Classification and Ranking," IEEE Access, vol. 9, pp. 123456–123470, 2021.
[14] "Fairness in AI-Based Hiring Systems: A Review of Bias and Mitigation Techniques," ACM Computing Surveys, vol. 54, no. 6, pp. 1–36, 2022.
[15] "Large Language Models for Information Retrieval: A Survey,", 2023.