Towards Explainable Job Title Matching: Leveraging Semantic Textual Relatedness and Knowledge Graphs

Vadim Zadykian, Bruno Andrade and Haithem Afli

ADAPT Centre, Munster Technological University, Cork, Ireland

Abstract

Semantic Textual Relatedness (STR) captures nuanced relationships between texts that extend beyond superficial lexical similarity. In this study, we investigate STR in the context of job title matching — a key challenge in resume recommendation systems, where overlapping terms are often limited or misleading. We introduce a self-supervised hybrid architecture that combines dense sentence embeddings with domain-specific Knowledge Graphs (KGs) to improve both semantic alignment and explainability. Unlike previous work that evaluated models on aggregate performance, our approach emphasizes data stratification by partitioning the STR score continuum into distinct regions: low, medium, and high semantic relatedness. This stratified evaluation enables a fine-grained analysis of model performance across semantically meaningful subspaces. We evaluate several embedding models, both with and without KG integration via graph neural networks. The results show that fine-tuned SBERT models augmented with KGs produce consistent improvements in the high-STR region, where the RMSE is reduced by 25% over strong baselines. Our findings highlight not only the benefits of combining KGs with text embeddings, but also the importance of regional performance analysis in understanding model behavior. This granular approach reveals strengths and weaknesses hidden by global metrics, and supports more targeted model selection for use in Human Resources (HR) systems and applications where fairness, explainability, and contextual matching are essential.

Keywords

STR, job title matching, knowledge graph, explainability, BERT, KG

1. Introduction

Semantic Textual Relatedness (STR) is a nuanced and context-dependent concept in Natural Language Processing (NLP) that measures the degree to which two text segments (words, sentences, or phrases) share semantically meaningful connections. Unlike Semantic Textual Similarity (STS) [1, 2, 3], which focuses primarily on surface-level closeness (e.g., synonyms or paraphrases), STR captures more abstract and associative relationships. For example, the words "mitten" and "glove" are semantically similar, whereas "hand" and "glove" are semantically related, yet dissimilar. STR also differs from Semantic Lexical Relatedness (SLR) [4, 5], which considers the relatedness of individual words rather than broader concepts.

While STS and SLR have been widely adopted in tasks such as paraphrase detection, question-answering, and summarization, STR remains underexplored, especially in domain-specific contexts such as talent acquisition and job recommender systems. In these settings, job titles often exhibit significant lexical diversity while denoting functionally similar or hierarchically related roles. Traditional keyword- or syntactic-based methods fail to account for this variation [6, 5]. For instance, "Chief Executive Officer" and "Managing Director" may have no shared tokens but represent nearly identical positions, whereas "Director of Sales" and "Vice President, Marketing" are distinct but related roles. This issue is compounded in global and multilingual hiring scenarios, where terminology is inconsistent or localized.

Another critical challenge in Human Resource (HR) applications is **explainability**. Job Recommender Systems (JRS) and Resume Recommender Systems (RRS) increasingly influence hiring decisions and career mobility. However, existing embedding-only methods such as Word2Vec [7] and BERT [8] often act as "black boxes," producing relatedness scores without justifications. This lack of transparency undermines user trust and makes compliance with regulatory requirements problematic.

To address these challenges, we propose a hybrid framework that combines fine-tuned Sentence-BERT (SBERT) [9] embeddings with domain-specific Knowledge Graphs (KGs). This integration leverages the complementary strengths of embeddings and KGs: embeddings capture contextual semantics even in the absence of lexical overlap, while KGs encode structured hierarchical and functional relationships (e.g., career progressions, job categories). Critically, KGs enable us to trace explicit reasoning paths behind predictions (e.g., "Project Lead" → "Team Leadership Roles" → "Program Manager"), thereby providing interpretability that is crucial in HR contexts. Our work offers the following contributions:

- we employ a self-supervised data pipeline that eliminates the need for manually labeled similarity scores by generating training pairs from cosine similarities between job descriptions
- we construct a knowledge graph to represent job-skill relationships and learn a neural mapping from textual embeddings to graph embeddings
- we introduce a hybrid modeling approach that integrates dense sentence embeddings with knowledge graph embeddings derived from a structured skill ontology
- we utilize a stratified evaluation framework by partitioning similarity scores into three interpretable regions: low, medium, and high STR. This region-aware analysis enables fine-grained assessment of model behavior that would otherwise be obscured by global metrics such as RMSE or Pearson correlation

We hypothesize that STR-aware systems augmented with KGs will produce more diverse and contextually relevant job matches, while also meeting the explainability needs of HR stakeholders. This positions STR and KGs not only as

RecSys in HR'25: The 5th Workshop on Recommender Systems for Human Resources, in conjunction with the 19th ACM Conference on Recommender Systems, September 22–26, 2025, Prague, Czech Republic.

vadim.zadykian@mymtu.ie (V. Zadykian); bruno.andrade@mtu.ie (B. Andrade); haithem.afli@mtu.ie (H. Afli)

0009-0009-9749-6332 (V. Zadykian); 0000-0002-1191-8082 (B. Andrade); 0000-0002-7449-4707 (H. Afli)

technical improvements but also as key enablers of fairness, trust, and transparency in modern workforce ecosystems.

2. Related Work

2.1. Semantic Textual Relatedness

Semantic Textual Relatedness (STR) is often confused with Semantic Textual Similarity (STS) [1]. STS can be considered a component of STR [10], as semantically similar texts (e.g., paraphrases) are inherently related. However, STR encompasses a broader range of semantic associations beyond surface-level resemblance, including hierarchical, causal, and contextual connections [11, 12, 6].

Abdalla et al. [10] demonstrate that integrating STR into search and recommendation pipelines enables retrieval of thematically relevant content, even when explicit term overlap is absent.

Recommender systems utilizing STR generally outperform those that rely solely on surface similarity [13]. These context-aware models can identify semantic connections between user preferences and items [14], improve personalization and diversity [15], and mitigate cold-start problems [16].

Recent advances in contextual language models have further propelled STR modeling. BERT [8] and Sentence-BERT (SBERT) [9] effectively capture polysemy and co-reference by leveraging bidirectional context.

Within the context of JRS and RRS, various BERT-based models are used for job title classification [17], resume classification [18], job-resume matching [19, 20, 21, 22], Named Entity Recognition [23, 24, 18], and semantic ranking of job recommendations [25, 26, 27].

In summary, STR has emerged as a key factor in the development of intelligent systems that require deep semantic understanding. Contextual language models, especially fine-tuned SBERT variants, provide a solid foundation for approximating STR [10, 28, 29, 30, 31].

2.2. Explainability in HR Systems

Transparency in HR systems is not only desirable, but it may also be mandatory. For example, the EU AI Act [32] classifies certain systems used in HR as "high-risk" and subjects them to strict traceability and explainability requirements. Explainability in HR systems, particularly in job recommendation systems, is indispensable for fostering fairness, transparency, loyalty and trust [33, 34, 35], facilitating informed decision-making [34], mitigating biases [36, 37], and catering to diverse stakeholder needs [38, 35, 39], while ensuring regulatory compliance.

2.3. Knowledge Graphs

Knowledge graphs (KGs) have emerged as valuable tools for enhancing the explainability of recommender systems.

Recent works have demonstrated how combining KGs with pre-trained language models (PLMs) — such as BERT and SBERT — can improve both knowledge graph completion and downstream semantic tasks [40, 41, 42].

Building on these insights, our approach leverages alignment between textual embeddings and structured knowledge graphs to improve job-to-job and job-to-skill similarity estimation. Inspired by multi-task frameworks Kim et al. [43] and contextualized reasoning models [44], we aim to train sentence encoders that not only capture linguistic nuance but are also sensitive to the structural topology of skills and industries.

2.4. Job Title Matching

Several recent studies have explored the use of representation learning and transformer-based models in the context of job matching and job title normalization. Zhang et al. [45] introduce Job2Vec, a multi-view framework that learns job title embeddings by integrating structured and unstructured job-related data. Lavi et al. [46] propose conSultantBERT, a fine-tuned Siamese SBERT model, which improves job–candidate matching over keyword-based approaches by leveraging domain-specific data. Building on similar ideas, Kaya and Bogers [47] investigate sentence-pair classification models using job titles as input signals and highlight the effectiveness of fine-tuned BERT embeddings for resume-to-job matching[20].

Decorte et al. [48] explore job title normalization using BERT-based models fine-tuned on recruitment corpora, showing improved classification into standardized taxonomies. Liu et al. [49] propose Title2Vec, a job title embedding approach designed for Named Entity Recognition (NER) and classification tasks. Zbib et al. [50] introduce a weakly supervised method for learning job title similarity by mining noisy skill co-occurrence patterns, offering a scalable alternative to manually labeled training data. In a related effort, Rosenberger et al. [22] present CareerBERT, a transformer-based architecture that aligns resumes and ESCO job descriptions for job recommendation tasks.

Collectively, these studies demonstrate the growing relevance of contextual and self-supervised embeddings in addressing real-world challenges in job matching, normalization, and recommendation. Our work builds upon these foundations and contributes further by integrating semantic representations with knowledge graph embeddings.

3. EXPERIMENTAL FRAMEWORK

3.1. Motivation and Research Objectives

Job Recommender Systems leverage a variety of input signals, including unstructured text (e.g., resumes or job postings), structured data (e.g., user preferences and skills), and collaborative features (e.g., interaction history between users and job listings). One commonly available yet often underutilized signal is the job title, which has been shown to carry meaningful semantic information [47].

While relying solely on job titles for matching would be insufficient, understanding the semantic textual relatedness (STR) between job titles can enhance filtering, refinement, and personalization in recommendation systems. This motivates our exploration of how job titles relate to one another and how these relationships can be explained —- contributing to more transparent and informed recommendations [34].

3.1.1. Research Objectives

- Develop a self-supervised approach for extracting semantic representations of skills and job functions.
- Automatically generate labeled datasets for training and evaluation.

- Assess text embedding strategies in capturing semantic relationships between job titles.
- Evaluate graph-based models for their ability to represent and explain job title relationships.
- Analyze model performance across the full STR spectrum.

3.2. Proposed Method

We propose a self-supervised pipeline for learning semantic representations of job titles and their alignment with skill knowledge graphs. The description of the pipeline is presented in Algorithm 1.

Algorithm 1 Self-Supervised Semantic Job Embedding and Skill Mapping Pipeline

Input: Raw job titles & descriptions, skill descriptions
Output: Trained job title to skill graph alignment model
Step 0: Summarization
    Apply a pretrained BART model [51] to summarize each job description, removing boilerplate and retaining functional content.
Step 1: Job Embedding Generation
    Encode each summary using SBERT to obtain auxiliary job embeddings.
Step 2: Pairwise Similarity Computation
    Compute cosine similarity between all job embeddings to generate relatedness scores.
Step 3: STR Dataset Construction and SBERT Fine-Tuning
    Construct a self-supervised dataset using similarity scores. Split into train/eval with disjoint job titles. Fine-tune SBERT on the training set.
Step 4: Skill Embedding Generation
    Encode textual descriptions of skills using a transformer model to obtain skill embeddings.
Step 5: Extraction of Job Functions and Skills
    For each job, compute cosine similarity to skills. Select top-ranked skills as semantic matches.
Step 6: Knowledge Graph Construction and Embedding
    Construct a bipartite graph of jobs and related skills. Learn node embeddings using a graph embedding model (e.g., RGCN, ComplEx).
Step 7: Embedding Alignment
    Train a neural network to map SBERT job title embeddings to the graph embedding space.
Return: Fine-tuned SBERT model and trained graph model.

Building upon insights from prior studies [43, 46, 47, 48, 49, 50, 42, 44, 22], our approach introduces several methodological innovations in the domain of job title similarity and normalization. We extend existing work by integrating self-supervised learning with pretrained language models and leveraging a skill-centric knowledge graph to enhance interpretability.

Departing from the Job2Vec framework [45], which relies on co-occurrence and structural signals, we adopt a self-supervised strategy that aligns contextual embeddings with knowledge graph embeddings built around explicit job-skill relationships. Unlike Lavi et al. [46] and Rosenberger et al. [22], whose models emphasize resume-job alignment or broad job matching, our focus is specifically on job title normalization and similarity estimation, leveraging both textual and structured graph-based representations.

While Decorte et al. [48] pursue supervised classification into a predefined taxonomy, our methodology emphasizes self-supervised learning techniques that aim to capture fine-grained semantic and structural relationships between job titles and skills. Similarly, although we share with Kaya and Bogers [47] the use of job titles as primary input signals, we extend this by incorporating richer semantic cues from full job descriptions. In comparison to Liu et al. [49], our model embeds a broader semantic scope by integrating additional contextual, relational, and graph-based information.

Finally, we align with the objective of Zbib et al. [50] in leveraging weak supervision for job title similarity, but diverge in our implementation by combining pre-trained sentence encoders with a structured knowledge graph of job-skill relationships. This enables our model to move beyond raw skill co-occurrence signals and instead produce semantically aligned, interpretable embeddings that support robust similarity estimation across diverse job roles.

3.3. Text Vectorization Models

We evaluate five different text vectorization model configurations which are summarized in Table 1.

Vectorizer	Model Description	Size
JOBBERT	Pre-trained JOBBERT model [48] based on the SBERT architecture 'jjzha/jobbert-base-cased'	768
JOBBERT-F	JOBBERT model 'jjzha/jobbert-base-cased', fine-tuned on synthetic data	768
MPNET	Pre-trained MPNET model based on the SBERT architecture 'all-mpnet-base-v2'.	768
MPNET-F	MPNET 'all-mpnet-base-v2', fine-tuned on synthetic data	768
MPNET+RGCN	Fine-tuned MPNET model 'all-mpnet-base-v2', coupled with a Graph R-GCN model [52]. R-GCN was selected because it is designed for node-level tasks (e.g., classification, regression).	500

Table 1
Vectorization models evaluated for the STR learning task

We evaluate every model on the same validation dataset which contains diverse job title pairs not included in any of the training data. For evaluation of STR prediction, we adopt Root Mean Squared Error (RMSE). This choice is valid because we want to optimize for precise similarity scores, and not just relative order.

3.4. Implementation Details

The experiments were carried out in the Google Colab environment [53] with specifications listed in Table 5 (see Appendix A). The proposed pipeline requires several key hyper-parameters which are specified in Table 4 (see Appendix A). The implementation code was written in Python [54] using Visual Studio [55]. The source code, together with the input and output files, is available at [56].

3.4.1. Mapping Text to KG Space

Training. We fine-tune the SBERT model using anchor-sample-score triplets and cosine similarity loss. We learn a parametric mapping that projects a job title's SBERT embedding into the knowledge-graph (KG) embedding space. We use a lightweight MLP with ℓ_2 normalization and MSE embedding loss.

Inference. At inference time, we encode job titles via fine-tuned SBERT to obtain text embedding vectors, then compute graph embedding vectors, and calculate cosine similarity to estimate the STR score.

3.4.2. Skill Selection and Graph Pruning

To prevent generic skills from dominating the graph and explanations, we remove skills with job share > 20%. Remaining skills are reweighted by a specificity score (inverse centrality degree). This yields a sparser, more discriminative KG and improves both retrieval quality and explanation specificity.

3.5. Stratified Data Sampling

To better understand and evaluate model behavior across varying levels of semantic textual relatedness (STR), we partition the continuous STR range into three semantically meaningful zones (Figure 1), guided by domain expertise in HR:

- Low STR (0.0–0.50): Pairs of job titles that are largely unrelated or noisy, often representing very different occupations or sectors.
- Medium STR (0.50–0.75): Ambiguous or borderline cases, which tend to be more challenging due to partial overlap in semantics.
- High STR (0.75–1.0): Highly related job title pairs, including near-duplicates, synonyms, or slight variations of the same role.

This stratification enables a more nuanced evaluation of model performance, as global performance metrics (e.g., overall RMSE or Pearson correlation) may obscure variation in model behavior across different semantic regions. We hypothesize that some models will perform better in specific STR regions while underperforming in others. For example, models trained with Cosine Similarity Loss may effectively distinguish highly similar and dissimilar titles but struggle with borderline cases.

3.6. Data

The raw data are obtained from several open-source collections: a Kaggle dataset [57], a list of granular skills and competencies by ESCO [58], and a list of broad job functions Indeed job site [59]. Table 6 (see Appendix A) describes the data files consumed and produced by the system.

4. Results & Discussion

Each vectorization model is evaluated on the same validation dataset where the predicted STR is calculated using the cosine distance between two final embedding vectors. We calculate the global RMSE as well as RMSE values for every data region of interest (i.e. Low STR, Medium STR, High STR), which are presented in Table 2.

We also perform paired t-tests to reveal how each model's performance varies across STR regions (Low, Medium, High) based on absolute errors (Table 3 and Figure 2).

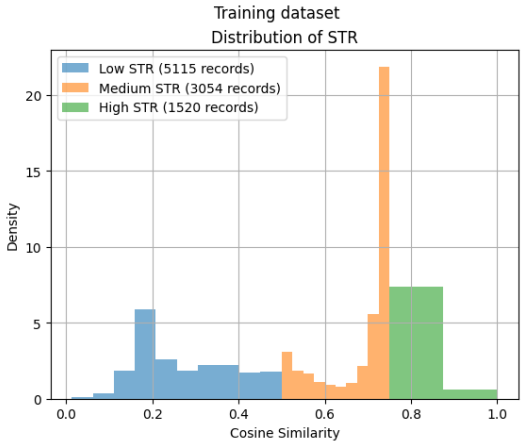

Figure 1: Distribution of STR in Training dataset

Vectorizer	Global STR	Low STR	Medium STR	High STR
JOBBERT	0.28	0.38	**0.11**	0.15
JOBBERT-F	0.17	0.16	0.17	0.18
MPNET	0.29	**0.14**	0.36	0.44
MPNET-F	**0.16**	**0.14**	0.16	0.18
MPNET+RGCN	0.23	0.30	0.14	**0.11**

Table 2
RMSE values across STR regions

Model	Comparison	t-value	p-value
JOBBERT	Low *vs* Medium	28.40	0.00
	Medium *vs* High	-4.61	0.00
	Low *vs* High	23.04	0.00
JOBBERT-F	Low *vs* Medium	-2.11	0.04
	Medium *vs* High	-0.12	0.90
	Low *vs* High	-2.11	0.04
MPNET	Low *vs* Medium	-19.38	0.00
	Medium *vs* High	-5.57	0.00
	Low *vs* High	-24.21	0.00
MPNET-F	Low *vs* Medium	-2.23	0.03
	Medium *vs* High	-0.90	0.37
	Low *vs* High	-3.03	0.00
MPNET-RGCN	Low *vs* Medium	20.43	0.00
	Medium *vs* High	3.72	0.00
	Low *vs* High	23.77	0.00

Table 3
Paired t-test results for absolute errors across STR regions

4.1. Evaluation of Region-Specific Model Behavior

Our analysis highlights the limitations of aggregate metrics such as global RMSE, which may mask meaningful variation in model performance across different STR zones, as demonstrated in Table 2. Stratified evaluation reveals that models often exhibit asymmetric performance as evident

from Figure 2. A model may perform reliably in distinguishing unrelated job titles (low STR), yet be less consistent in matching similar roles (medium to high STR).

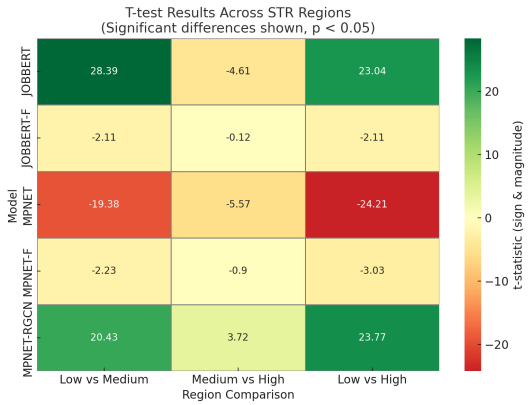

Figure 2: T-Test Results Across STR Regions

The paired t-test analysis reveals statistically significant differences in model performance across STR regions, as measured by absolute errors.

4.1.1. JOBBERT

JOBBERT demonstrates positive t-statistics in "Low vs Medium" and "Low vs High" which suggests that the model performs better in Medium and High STR bands compared to Low. Conversely, the negative t-statistic in "Medium vs High" implies superior performance in the Medium band (Figure 3).

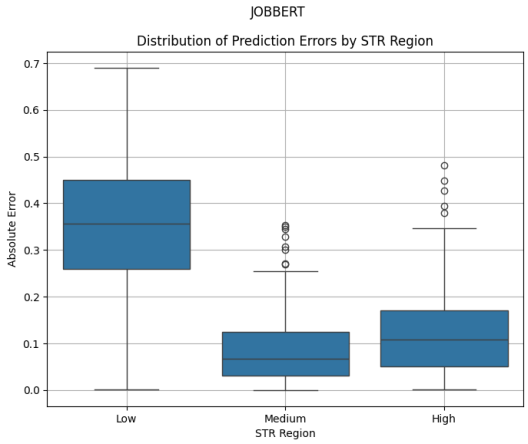

Figure 3: Distribution of Prediction Errors for JOBBERT

4.1.2. JOBBERT-F

JOBBERT-F exhibits fewer significant differences: low–medium and low–high contrasts are significant, but medium–high differences are not. This stability in medium and high STR regions may reflect the benefits of fine-tuning in reducing variance for more semantically similar pairs (Figure 4).

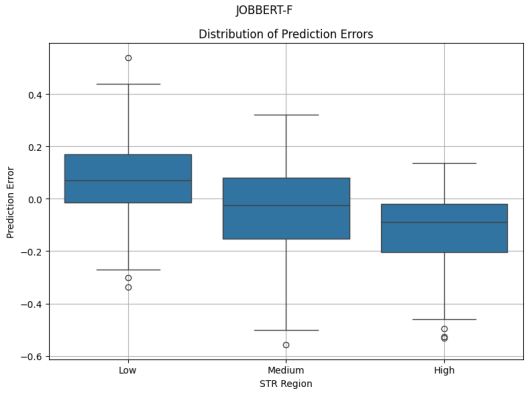

Figure 4: Distribution of Prediction Errors for JOBBERT-F

4.1.3. MPNET

MPNET shows strong, significant differences across all region pairs, with particularly large negative t-values for low–medium and low–high comparisons, indicating much lower errors in low STR compared to the other regions (Figure 5).

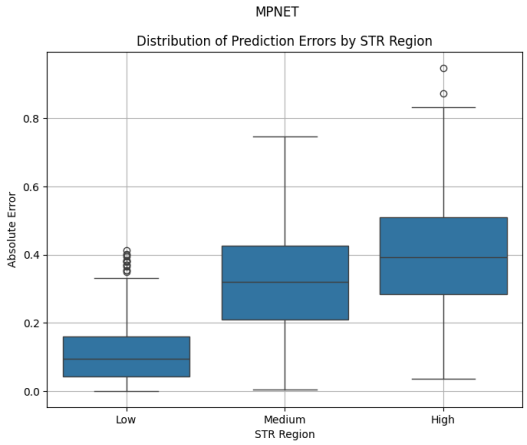

Figure 5: Distribution of Prediction Errors for MPNET

4.1.4. MPNET-F

MPNET-F shows strong, significant differences across all region pairs, with particularly large negative t-values for low–medium and low–high comparisons. However, it shows no significant difference between medium and high STR, suggesting more consistent performance at the higher end of the similarity spectrum (Figure 6).

4.1.5. MPNET-RGCN

MPNET-RGCN demonstrates significant differences for all region pairs, with large positive t-values, implying higher errors in low STR and substantially lower errors in medium and high STR regions. This suggests that KG integration particularly benefits the model's handling of semantically similar job pairs (Figure 7).

Overall, these results confirm our hypothesis that models behave differently across STR regions, and that fine-tuning

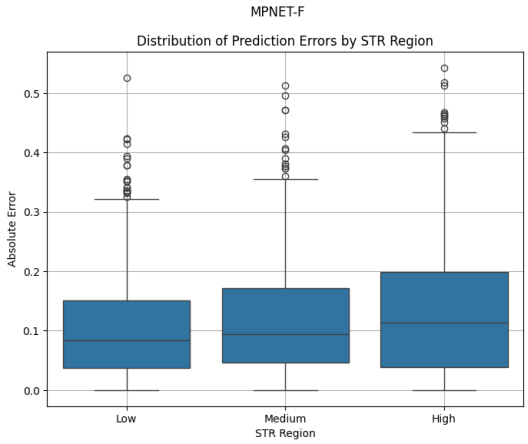

Figure 6: Distribution of Prediction Errors for MPNET-F

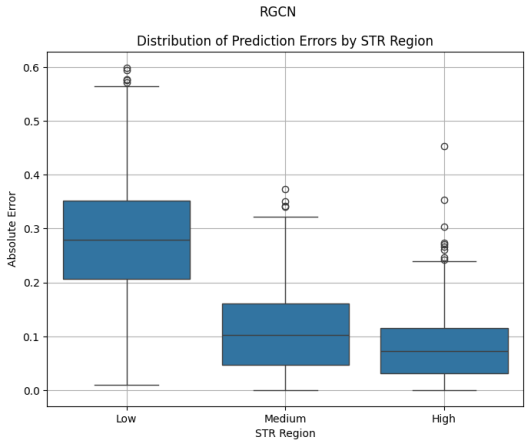

Figure 7: Distribution of Prediction Errors for MPNET-RGCN

or KG integration can improve performance stability in specific ranges.

4.2. Explainability Analysis

A key motivation of our work is to provide **explainable** job-to-job matches through the integration of knowledge graphs, linking jobs and skills, and providing structured explanations.

Figures 8 and 9 present two illustrative cases: one good match and one poor match. For the high-STR pair *"Senior Performance and Project Analyst"* vs. *"Director, eCommerce & Retail"*, the explanation highlights a shared set of highly-specific skills (e.g., *"supervise brand management"* with specificity of 0.67). In contrast, the low-quality match *"Executive Office Assistant"* vs. *"Help Desk Shift Supervisor* is driven by overly generic skills (e.g., *"supervise office workers"* with specificity of 0.0). Such explanations increase transparency by showing whether similarity arises from meaningful or spurious overlaps.

These results strengthen the claim that KGs improve explainability. Good matches can be justified by showing the specific shared skills that drive similarity, while poor matches can be diagnosed by revealing over-reliance on generic skills.

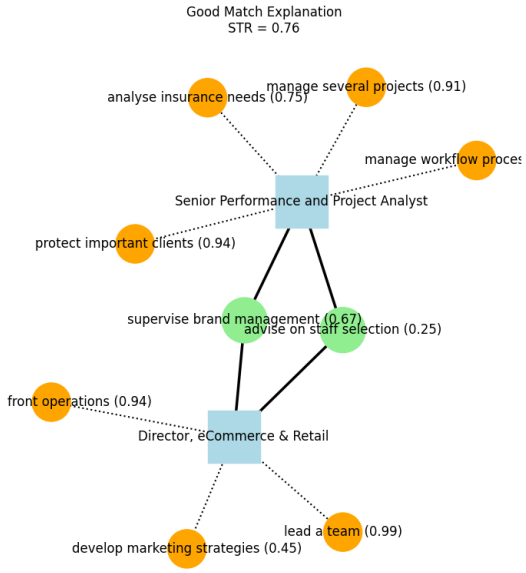

Figure 8: Explanation Graph - Good Job Title match. The skill specificity (shown in brackets) is the inverse of the centrality degree

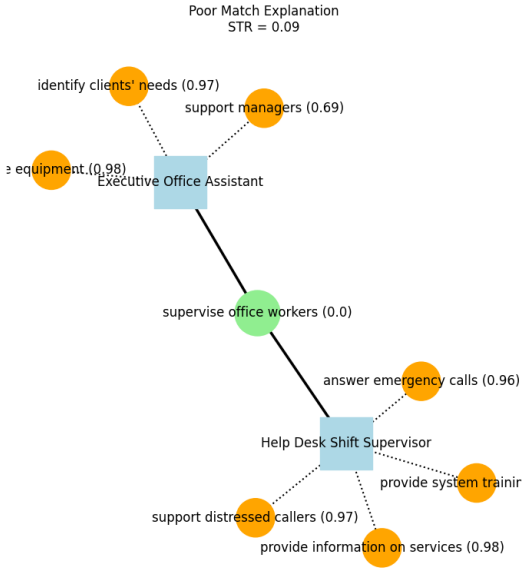

Figure 9: Explanation Graph - Poor Job Title match. The skill specificity (shown in brackets) is the inverse of the centrality degree

4.3. Practical Implications

Our findings have important practical implications for downstream tasks such as re-ranking or candidate filtering. When irrelevant matches have already been discarded, the primary challenge lies in fine-grained differentiation among relevant alternatives. In such contexts, a model's behavior in the medium to high STR range becomes especially critical (Figure 7). Conversely, when filtering for dissimilar job pairs (e.g., in deduplication or anomaly detection tasks), performance in the low STR range is of greater interest (Figure 5). This suggests that a general-purpose pre-trained Language Model can be used in the initial stages of the

recommendation pipeline, with a transition to fine-tuned, domain-specific models at a later stage when more detailed distinctions are necessary.

Stratified evaluation also allows us to observe localized performance patterns and better characterize where models succeed or fail. This approach informs both model selection and training strategies — such as adapting loss functions to focus on under-performing regions or augmenting training data with examples from underrepresented STR bands.

Furthermore, by integrating Knowledge Graphs into the semantic matching process, we provide a structured reasoning path behind recommendations. This may improve transparency, build trust with stakeholders, and help recruiters justify why a specific job was recommended.

Although focused on job title and skill matching, our methodology can be applied to other domains such as academic paper recommendation, product matching, or legal case retrieval.

4.4. Future Work

Although our current approach provides a foundation for learning semantic representations of job titles using graph-based and textual signals, there are opportunities for future improvement and expansion.

First, our knowledge graph construction is limited to skill-based relationships. Incorporating additional semantic dimensions such as industry classifications, job seniority levels (e.g., "Lead Engineer" vs. "Intern"), and domain-specific contexts (e.g., "Data Scientist – Healthcare" vs. "Data Scientist – Finance") would provide a more comprehensive representation of the job landscape.

Second, the scope of our current model evaluation is constrained. We only explore a limited set of graph embedding models, focus exclusively on Cosine Similarity Loss, and implement a single negative sampling strategy. Future research should embrace a broader range of models, loss functions (e.g., contrastive loss, triplet loss), and negative sampling strategies to assess their effect on model performance.

Third, our work focuses exclusively on job-to-job matching. Extending the framework to job-to-resume and resume-to-job matching tasks could make it more useful in real-world recruitment systems.

Additionally, our approach currently relies on structured skill taxonomies such as ESCO [58] or O*NET [60], which may limit generalizability in domains with less formalized ontologies. Also, we do not address multi-lingual job descriptions, which presents an important direction for future development, particularly for global labor markets.

Moreover, our reliance on weak supervision introduces potential label noise, especially in low-similarity cases, which raises concerns about general robustness and may limit the model's ability to generalize.

Finally, we train and evaluate on a relatively small dataset, which may not capture the full variability of job title semantics across sectors or regions. Scaling the data set and introducing data augmentation techniques or semi-supervised learning methods could mitigate this limitation and improve the robustness of the model.

5. Conclusion

This study sets the direction for addressing a critical challenge in HR applications: the need for explainable and transparent recommendations. Although text embedding models like SBERT capture complex contextual semantics, they often lack interpretability. To overcome this limitation, we introduce a hybrid approach that combines Semantic Textual Relatedness (expressed by fine-tuned SBERT embeddings) with domain-specific knowledge graphs using Graph Neural Networks. This combination enables not only enhanced performance but also the ability to trace reasoning paths between matched job titles —- an essential feature for auditable and trustworthy decision-making in hiring contexts. As the use of Artificial Intelligence in recruitment systems expands, approaches that prioritize both semantic depth and interpretability will be key to ensuring fairness and user trust.

Ultimately, this research contributes to building intelligent, equitable and explainable recommendation systems that serve both candidates and employers in a dynamic and evolving labor market.

Acknowledgments

This research was partially supported by the Horizon Europe project GenDAI (Grant Agreement ID: 101182801) and by the ADAPT Research Centre at Munster Technological University. ADAPT is funded by Taighde Éireann – Research Ireland through the Research Centres Programme and co-funded under the European Regional Development Fund (ERDF) via Grant 13/RC/2106_P2.

We would also like to thank the anonymous reviewers for their valuable feedback and constructive suggestions, which have helped to improve the quality and clarity of this work.

Declaration on Generative AI

In our literature review process, we employed Scite AI Research Assistant [61], which allowed a more comprehensive review of the existing literature, ensuring that the sources included in this study were relevant and reliable. We utilized OpenAI Code Assistant [62] to accelerate code development and improve productivity during the implementation phase. The assistant was used to generate boilerplate code, troubleshoot and debug runtime errors, and explore alternative design patterns. We also used Grammarly [63] and Microsoft Copilot [64] for paraphrasing and corrections of grammatical, syntactical, and other writing errors. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. Generative AI tools were not used for data analysis, experimentation, or the formulation of hypotheses and conclusions.

References

[1] D. Chandrasekaran, V. Mago, Evolution of semantic similarity—a survey, ACM Comput. Surv. 54 (2021). URL: https://doi-org.mtu.idm.oclc.org/10.1145/3440755. doi:10.1145/3440755.

[2] J. Chen, Q. Huang, C. Wang, C. Li, Sensemap: Urban performance visualization and analytics via semantic textual similarity, IEEE Transactions on Visualization and Computer Graphics 30 (2024) 6275–6290. doi:10.1109/TVCG.2023.3333356.

[3] R. Gaur, A. Dwivedi, Knowledge graph-based evaluation metric for conversational ai systems: A step towards quantifying semantic textual similarity, in: S. Dhar, S. Goswami, U. Dinesh Kumar, I. Bose, R. Dubey, C. Mazumdar (Eds.), AGC 2023, Springer Nature Switzerland, Cham, 2024, pp. 112–124.

[4] D. A. Cruse, Lexical semantics, Cambridge university press, 1986.

[5] S. Webb, Word families and lemmas, not a real dilemma: Investigating lexical units, Studies in Second Language Acquisition 43 (2021) 973–984. doi:10.1017/S0272263121000760.

[6] Š. Zikánová, E. Hajičová, B. Vidová-Hladká, P. Jínová, J. Mírovskỳ, A. Nědolužko, L. Poláková, K. Rysová, M. Rysová, J. Václ, Discourse and coherence: from the sentence structure to relations in text, Ústav formální a aplikované lingvistiky, 2015.

[7] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013). URL: https://arxiv.org/pdf/1301.3781.

[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/V1/N19-1423.

[9] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019). doi:10.18653/v1/d19-1410.

[10] M. A. Abdalla, K. Vishnubhotla, S. M. Mohammad, What makes sentences semantically related: a textual relatedness dataset and empirical study (2021). doi:10.48550/arxiv.2110.04845.

[11] G. Sloan, Relational ambiguity between sentences, College Composition and Communication 39 (1988) 154–165. URL: http://www.jstor.org/stable/358025.

[12] Y. Miyabe, H. Takamura, M. Okumura, Identifying cross-document relations between sentences, in: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I, 2008.

[13] P. Lombardo, A. Boiardi, L. Colombo, A. Schiavone, N. Tamagnone, Top-rank-focused adaptive vote collection for the evaluation of domain-specific semantic models, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 3081–3093. URL: https://aclanthology.org/2020.emnlp-main.249/. doi:10.18653/v1/2020.emnlp-main.249.

[14] M. Ye, Z. Tang, J. Xu, L. Jin, Recommender system for e-learning based on semantic relatedness of concepts, Information 6 (2015) 443–453. doi:10.3390/info6030443.

[15] S. Likavec, F. Osborne, F. Cena, Property-based semantic similarity and relatedness for improving recommendation accuracy and diversity, International Journal on Semantic Web and Information Systems (IJSWIS) 11 (2015) 1–40. doi:10.4018/ijswis.2015100101.

[16] S. Natarajan, S. Vairavasundaram, K. Kotecha, V. Indragandhi, S. Palani, J. R. Saini, L. Ravi, Cd-semmf: Cross-domain semantic relatedness based matrix factorization model enabled with linked open data for user cold start issue, IEEE Access 10 (2022) 52955–52970. doi:10.1109/ACCESS.2022.3175566.

[17] I. Rahhal, K. M. Carley, I. Kassou, M. Ghogho, Two stage job title identification system for online job advertisements, IEEE Access 11 (2023) 19073–19092. doi:10.1109/ACCESS.2023.3247866.

[18] S. Tanberk, S. S. Helli, E. Kesim, S. N. Cavsak, Resume matching framework via ranking and sorting using nlp and deep learning, in: 2023 8th International Conference on Computer Science and Engineering (UBMK), 2023, pp. 453–458. doi:10.1109/UBMK59864.2023.10286605.

[19] S. A. Pias, M. Hossain, H. Rahman, M. M. Hossain, Enhancing job matching through natural language processing: A bert-based approach, in: 2024 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2024, pp. 1–6. doi:10.1109/ICISET62123.2024.10939860.

[20] M. Kaya, T. Bogers, An exploration of sentence-pair classification for algorithmic recruiting, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1175–1179. URL: https://doi-org.mtu.idm.oclc.org/10.1145/3604915.3610657. doi:10.1145/3604915.3610657.

[21] S. Rezaeipourfarsangi, E. E. Milios, Ai-powered resume-job matching: A document ranking approach using deep neural networks, in: Proceedings of the ACM Symposium on Document Engineering 2023, DocEng '23, Association for Computing Machinery, New York, NY, USA, 2023. URL: https://doi-org.mtu.idm.oclc.org/10.1145/3573128.3609347. doi:10.1145/3573128.3609347.

[22] J. Rosenberger, L. Wolfrum, S. Weinzierl, M. Kraus, P. Zschech, Careerbert: Matching resumes to esco jobs in a shared embedding space for generic job recommendations, Expert Systems with Applications (2025) 127043. URL: https://www.proquest.com/scholarly-journals/careerbert-matching-resumes-esco-jobs-shared/docview/3213850804/se-2.

[23] G. L, R. M L, G. H B, K. Mathada, M. B, Intelligent resume scrutiny using named entity recognition with bert, in: 2023 International Conference on Data Science and Network Security (ICDSNS), 2023, pp. 01–08. doi:10.1109/ICDSNS58469.2023.10245304.

[24] P. Dhobale, V. Bhoir, A. Vyavhare, P. Yelkar, P. A. Dharmadhikari, Resumatcher: An intelligent resume ranking system, in: 2025 3rd International Confer-

ence on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2025, pp. 1778–1783. doi:10.1109/IDCIOT64235.2025.10915179.

[25] J. Tang, H. Chen, Z. Chen, J. Pan, Y. He, J. Zhao, Z. Li, A person-job matching method based on bm25 and pre-trained language model, in: Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing, MLNLP '23, Association for Computing Machinery, New York, NY, USA, 2024, p. 78–83. URL: https://doi-org.mtu.idm.oclc.org/10.1145/3639479.3639494. doi:10.1145/3639479.3639494.

[26] R. Ramyar, G. Nagarani, S. Natarajan, Deep learning based approach to streamline resume categorization and ranking, in: 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS), 2024, pp. 840–845. doi:10.1109/ICICNIS64247.2024.10823187.

[27] M. A. Aleisa, N. Beloff, M. White, Implementing airm: a new ai recruiting model for the saudi arabia labour market, Journal of Innovation and Entrepreneurship 12 (2023) 59. URL: https://www.proquest.com/scholarly-journals/implementing-airm-new-ai-recruiting-model-saudi/docview/2864704180/se-2.

[28] M. Zhao, P. Dufter, Y. Yaghoobzadeh, H. Schütze, Quantifying the contextualization of word representations with semantic class probing, Findings of the Association for Computational Linguistics: EMNLP 2020 (2020). doi:10.18653/v1/2020.findings-emnlp.109.

[29] K. Misra, A. Ettinger, J. T. Rayz, Exploring bert's sensitivity to lexical cues using tests from semantic priming, Findings of the Association for Computational Linguistics: EMNLP 2020 (2020) 4625–4635. doi:10.18653/v1/2020.findings-emnlp.415.

[30] A. Zou, Z. Zhang, H. Zhao, Eventbert: incorporating event-based semantics for natural language understanding, Lecture Notes in Computer Science (2022) 66–80. doi:10.1007/978-3-031-18315-7_5.

[31] A. Hammami, S. B. Abdallah Ben Lamine, H. Baazaoui, Bert-based semantic relations extraction from large-scale medical datasets, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 6460–6468. doi:10.1109/BigData62323.2024.10825162.

[32] Regulation (eu) 2024/1689 of the european parliament and of the council of 12 july 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act), Official Journal of the European Union, L 168, 12 July 2024, p. 1–157, 2024. URL: https://artificialintelligenceact.eu/, applies from 2 February 2025.

[33] H. Min, B. Yang, D. G. Allen, A. A. Grandey, M. Liu, Wisdom from the crowd: can recommender systems predict employee turnover and its destinations?, Personnel Psychology 77 (2022) 475–496. doi:10.1111/peps.12551.

[34] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, Foundations and Trends® in Information Retrieval (2020). doi:10.1561/1500000066.

[35] G. Bied, S. Nathan, E. Perennes, M. Hoffmann, P. Caillou, B. Crépon, C. Gaillac, M. Sebag, Toward Job Recommendation for All, in: IJCAI 2023 - The 32nd International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Macau, China, 2023, pp. 5906–5914. URL: https://hal.science/hal-04245528. doi:10.24963/ijcai.2023/655.

[36] L. D. Chao Yang, Intelligent talent recommendation algorithm for college students for the future job market, Jes (2024). doi:10.52783/jes.1721.

[37] J. Yao, Y. Xu, J. Gao, A study of reciprocal job recommendation for college graduates integrating semantic keyword matching and social networking, Applied Sciences (2023). doi:10.3390/app132212305.

[38] R. Schellingerhout, Explainable multi-stakeholder job recommender systems (2024). doi:10.1145/3640457.3688014.

[39] R. Schellingerhout, F. Barile, N. Tintarev, A co-design study for multi-stakeholder job recommender system explanations (2023). doi:10.1007/978-3-031-44067-0_30.

[40] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, 2019. URL: https://arxiv.org/abs/1909.03193. arXiv:1909.03193.

[41] X. Liu, H. Hussain, H. Razouk, R. Kern, Effective use of bert in graph embeddings for sparse knowledge graph completion, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, SAC '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 799–802. URL: https://doi.org/10.1145/3477314.3507031. doi:10.1145/3477314.3507031.

[42] Y. Xu, Q. Su, Boosting bert-based knowledge graph completion with contrastive learning and hard sample training, Procedia Computer Science 222 (2023) 71–80. URL: https://www.sciencedirect.com/science/article/pii/S1877050923009109. doi:https://doi.org/10.1016/j.procs.2023.08.145, international Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023).

[43] B. Kim, T. Hong, Y. Ko, J. Seo, Multi-task learning for knowledge graph completion with pre-trained language models, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1737–1743. URL: https://aclanthology.org/2020.coling-main.153/. doi:10.18653/v1/2020.coling-main.153.

[44] H. Gul, A. G. Naim, A. A. Bhat, A contextualized bert model for knowledge graph completion, 2024. URL: https://arxiv.org/abs/2412.11016. arXiv:2412.11016.

[45] D. Zhang, J. Liu, H. Zhu, Y. Liu, L. Wang, P. Wang, H. Xiong, Job2vec: Job title benchmarking with collective multi-view representation learning, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2763–2771. URL: https://doi.org/10.1145/3357384.3357825. doi:10.1145/3357384.3357825.

[46] D. Lavi, V. Medentsiy, D. Graus, consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers, 2021. URL: https://arxiv.org/abs/2109.06501. arXiv:2109.06501.

[47] M. Kaya, T. Bogers, Effectiveness of job title based embeddings on résumé to job ad recommendation, in: CEUR Workshop Proceedings, volume 2967, CEUR Workshop Proceedings, 2021. URL: https://vbn.aau.dk/

ws/portalfiles/portal/464971521/Kaya_Bogers.pdf.
[48] J.-J. Decorte, J. Van Hautte, T. Demeester, C. Develder, Jobbert: Understanding job titles through skills, 2021. URL: https://www.proquest.com/working-papers/jobbert-understanding-job-titles-through-skills/docview/2574960980/se-2.
[49] J. Liu, Y. C. Ng, Z. Gui, T. Singhal, L. Blessing, K. L. Wood, K. H. Lim, Title2vec: a contextual job title embedding for occupational named entity recognition and other applications, Journal of Big Data 9 (2022). doi:10.1186/s40537-022-00649-5.
[50] R. Zbib, A. L. Lucas, F. Retyk, R. Poves, J. Aizpuru, H. Fabregat, V. Simkus, E. García-Casademont, Learning job titles similarity from noisy skill labels, 2023. doi:10.48550/arxiv.2207.00494.
[51] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.
[52] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, 2017. URL: https://arxiv.org/abs/1703.06103. arXiv:1703.06103.
[53] Google, Google colaboratory, https://colab.research.google.com/, 2024. Accessed: 2025-08-02.
[54] Python Core Team, Python: A dynamic, open source programming language, Python Software Foundation, 2019. URL: https://www.python.org/.
[55] Microsoft Corporation, Visual studio, https://visualstudio.microsoft.com/, 2025. Integrated development environment (IDE) by Microsoft. Accessed: 2025-08-07.
[56] V. Zadykian, Job title relatedness, ????
[57] A. Koneru, Linkedin job postings (2023 - 2024), 2024. doi:10.34740/KAGGLE/DSV/9200871.
[58] E. Commission, S. A. Directorate-General for Employment, Inclusion, ESCO handbook – European skills, competences, qualifications and occupations, Publications Office of the European Union, 2019. doi:doi/10.2767/451182.
[59] I. E. Team, 230 job titles in 17 industries to include on your resume, 2025. URL: https://www.indeed.com/career-advice/resumes-cover-letters/job-title.
[60] U.S. Department of Labor, O*net online, 2025. URL: https://www.onetonline.org/, accessed: 2025-08-06.
[61] Scite, Inc., Ai research assistant, 2024. URL: https://scite.ai/assistant.
[62] OpenAI, Openai code assistant, 2025. URL: https://openai.com, large language model used for code generation and assistance.
[63] Grammarly, Grammarly grammar checker, n.d. URL: https://www.grammarly.com/grammar-check, accessed: 2025-04-21.
[64] Microsoft, Copilot (gpt-4) [large language model], https://copilot.microsoft.com/, 2025. Accessed: 2025-08-07.

6. Appendices

A. Tables

Parameter	Description	Value
High STR Range	Data region in which STR is considered 'high'	[0.75, 1.0]
Medium STR Range	Data region in which STR is considered 'medium'	[0.5, 0.75]
Low STR Range	Data region in which STR is considered 'low'	[0, 0.5]
Number of Skills Per Job	Maximum number of skills assigned to a job	10
Job-Skill STR Threshold	Minimum STR value at which a skill is considered related to a job	0.5
Skill-Skill STR Threshold	Minimum STR value at which a child skill is considered related to a parent skill	0.25
Text Epochs	Number of epochs for SBERT fine-tuning	5
Graph Epochs	Number of epochs for KG model training	15

Table 4
Hyper-parameters of the proposed pipeline

Name	Description
Environment	Google Colab
Architecture	x86/x64
CPU	8 CPUs @ 2.00GHz
GPU	Tesla T4
CUDA	Version 12.4

Table 5
Runtime Environment Setup

File Name	File Description
Source Jobs	**Input**. File 'source_jobs.csv' is derived from a Kaggle dataset [57] and contains 14,000 records covering a wide range of professional roles: technical, creative, educational, financial, administrative, and operational.
Source Skills	**Input**. File 'source_skills.csv' is derived from a list of 14,000 skills and competences [58] and a list of 50 skill categories [59].
Source Skill Hierarchy	**Input**. File 'source_skill_hierarchy.csv' defines relationships between skill categories. For example, 'Marketing' and 'Sales' are grouped under 'Sales & Marketing'
Training Job Title Pairs	**Output**. File 'train_job_title_pairs.csv' is a training dataset. Each row includes anchor text (job title) which is the basis for comparison, sample text (job title), and STR score between the anchor and the sample
Evaluation Job Title Pairs	**Output**. File 'eval_job_title_pairs.csv' is an evaluation dataset. Each row includes anchor text (job title) which is the basis for comparison, sample text (job title), and STR score between the anchor and the sample

Table 6
Description of main data files