Graph-Based Job Recommendation System

Leveraging Network Structures and Embeddings for Enhanced Employment Matching in Singapore

Amir Yunus
Nanyang Technological University
Singapore
G2404774J
W240004@e.ntu.edu.sg

Brandon Gay
Nanyang Technological University
Singapore
G2405389J
pengrend001@e.ntu.edu.sg

Lee Oon Teng
Nanyang Technological University
Singapore
G2404900F
oonteng001@e.ntu.edu.sg

***Abstract*—This paper presents a graph-based job recommendation system for Singapore's employment market. By leveraging Graph Neural Networks (GNNs) and GraphSAGE architecture, the system transforms approximately 50,000 job listings into a structured graph network with 25,142 nodes and 79 million edges. The system incorporates multiple relationship types, including company affiliation, job similarity, and geographic proximity, while utilising transformer-based embeddings for semantic analysis. Our implementation achieves efficient real-time recommendations through optimised caching mechanisms and approximate nearest neighbour search, demonstrating strong performance in both structural and semantic dimensions. The system's architecture supports dynamic updates and includes an intuitive user interface for interaction with the underlying graph structure. Experimental results show effective capture of explicit and latent job market relationships, though limitations in spatial distribution handling and scalability are noted for future research.**

I. Introduction

In Singapore's competitive job market, job seekers often face an overwhelming number of online job postings with diverse requirements and descriptions. This high volume, combined with the city's dynamic economic landscape, makes it challenging for individuals to find roles that match their qualifications and preferences. Current keyword-based search methods struggle to capture the nuances of job listings and often provide inadequate results [1].

To tackle this issue, we are developing a one-click job recommender system specifically tailored to Singapore's job market. Using advanced machine learning, particularly Graph Neural Networks (GNNs) and collaborative filtering, our system aims to simplify job searching. Unlike traditional search engines that require extensive manual filtering, our approach models job listings as connected nodes in a network, capturing both job content and relationships such as skill overlap, industry similarities, and geographic proximity [2].

The project presents technical challenges, as our dataset includes around 50,000 raw job entries with inconsistent data formats. This diversity in address formats, job descriptions, and industry classifications requires extensive data cleaning and structuring to make it usable. Singapore's varied job landscape necessitates a model capable of recognising subtle relationships between job attributes that go beyond basic keyword filtering [3].

By structuring the data into nodes and edges, our system captures complex relationships specific to Singapore's job market. Using embeddings derived from job descriptions, the model can understand and recommend jobs that align with user profiles, prioritising precision and ease of use. This design, leveraging graph-based modelling and natural language processing, creates a robust job recommender system that provides data-driven, personalised suggestions to job seekers [4].

II. Related Work

A. Evolution of Job Recommendation Systems

Job recommendation systems have evolved significantly with advancements in machine learning and increasing labour market complexity [6]. Early systems relied on keyword matching and basic collaborative filtering, which were limited in capturing the nuanced relationships between jobs and candidates and suffered from cold-start issues [7] [8]. Content-based filtering, using NLP techniques like TF-IDF, improved accuracy but couldn't adapt to changing job market dynamics [9].

Modern systems now employ deep learning, including transformers and attention mechanisms, to better understand semantic relationships in job data and capture latent job attributes [10]. Graph-based models mark a paradigm shift by mapping the job market as a network, enabling Graph Neural Networks (GNNs) to learn both job attributes and structural relationships effectively [2].

B. Graph-Based Recommendation Models

Graph-based models have transformed job recommendation by capturing complex relationships in job markets, representing job listings, skills, and industry links as interconnected nodes and edges [10]. Graph Neural Networks (GNNs) are especially effective in aggregating information from connected nodes, identifying both local and global patterns that traditional models might miss [11] [12].

Graph Attention Networks (GATs) enhance this by weighting node relationships, allowing the model to prioritise relevant job connections in varied recommendation contexts [13]. New techniques like GraphSAGE further allow GNNs to generalise to unseen nodes, which is crucial for recommending new job listings without retraining [3] [14].

Hybrid models that merge graph insights with content-based features from job descriptions create more accurate recommendations, integrating both explicit and implicit job relationships [15].

C. Limitations and Innovations

Despite significant advancements in job recommendation systems, limitations persist, particularly regarding personalisation and the integration of location-based data [16].

Traditional models, even when hybridised, often struggle to capture individual preferences and contextual features such as geographic constraints [14]. Location is especially critical in a city-state like Singapore, where commute times and location-specific preferences play a substantial role in job seeker decisions. Current graph-based models lack robust location-aware recommendations, highlighting an opportunity for improvement.

This project addresses these gaps by structuring scraped job data into a graph model, including embeddings to encode semantic job information and integrating geocoding for location-based considerations. By transforming raw, unstructured job data into a structured graph, our approach provides a higher level of personalisation and precision than traditional methods. We leverage a combination of GNN architectures and embedding techniques to capture job attributes, user preferences, and spatial data, producing a recommender system optimised for Singapore's distinct job market.

III. Task Description

A. Recommender System Task Definition

The primary goal of this research is to build a job recommendation system that leverages graph-based architectures to match job seekers with relevant job opportunities by analysing both structured and unstructured job data. Representing the job market as a complex network enables us to capture nuanced relationships between job positions, skills, company hierarchies, and geographic constraints, which are especially relevant in a densely connected market like Singapore.

Core System Components

The system architecture consists of three main components that work together to deliver personalised job recommendations:

1. Data Processing Pipeline: The data processing pipeline transforms scraped job listings by cleaning text columns for consistency. For unstructured text fields, such as job titles and descriptions, we summarise the descriptions before converting them into embeddings. This embedding process creates dense vector representations that capture semantic relationships within job data.
2. Graph Construction: We construct a heterogeneous graph where nodes represent individual job listings, and edges capture multiple types of relationships (e.g., skill similarity, company affiliation, and location proximity). This hybrid graph incorporates both structural information (through edge types and node features) and semantic data, allowing us to leverage cosine similarity for content matching and Gaussian kernel functions for calculating geographic proximity. Such multi-relational edges enable the model to identify patterns in job roles, skill clusters, and location-based preferences.
3. Recommendation Engine: At the core of the system is the recommendation engine, which utilises a GraphSAGE architecture with a mean pooling aggregation operation (aggr='mean'). This layer aggregates neighbouring node features by averaging them, producing node embeddings without applying attention weighting. This approach allows the model to learn representations based on the general neighbourhood structure without assigning specific importance to individual node relationships [3]. The system incorporates ranking signals like degree centrality and PageRank to further refine recommendations based on user preferences and geographic considerations [14].

Task Definition and Recommendation Process

The recommendation task is formulated as a link prediction problem, where the system predicts the likelihood of a connection between a user's preferences and specific job listings. This approach allows the system to recognise both local and global patterns across the graph and improve computational efficiency via neighbourhood sampling.

B. Dataset Overview

Our dataset, specifically curated for the Singapore job market, contains approximately 50,000 job listings with essential attributes, including job title, company name, job type, location, remote work status, and job descriptions. This diverse dataset is designed to support both content-based filtering and advanced graph-based analysis, providing a comprehensive foundation for capturing relationships within the job market [16].

Each job listing is represented as a node within a graph structure, and relationships between listings are represented by edges. These relationships capture important connections such as shared skills, company affiliation, and geographic proximity. Including spatial information through geocoding allows the model to recognise and incorporate location-based patterns critical for job seekers in Singapore, where commute time and proximity to work locations can significantly influence job decisions.

Unstructured fields, like job descriptions, are processed using a Large Language Model (LLM) for summarisation, followed by embedding conversion. This approach produces dense vector representations that enhance semantic similarity detection, which is especially useful in identifying related roles and industries that might not be evident through keywords alone. The structured data combined with these embeddings provides the graph-based recommendation model with a robust understanding of both explicit and latent job attributes, supporting nuanced recommendations.

C. Selection of Attributes and Rationale

The features selected for this job recommendation system are carefully chosen to capture core attributes that influence job matching while also ensuring computational efficiency. These features include both structural and semantic attributes, each contributing to a nuanced understanding of job listings and enabling the model to make personalised recommendations. The core features are as follows:

1. Job Title and Company: Job title and company name are fundamental for identifying relevant positions and employer preferences. Similar job titles or affiliations with certain companies provide essential

indicators of job relevance, supporting content-based filtering and clustering within the graph [2].

2. Job Type and Remote Work Status: Attributes specifying job type (e.g., full-time, part-time, contract) and remote work options are increasingly critical in the post-pandemic job market. These features directly influence the graph's structure and provide filtering options aligned with user preferences, which is crucial for Singapore's job market, where remote and hybrid work arrangements are growing in demand [16].
3. Location: Including geographic coordinates for each job listing enables the system to recognise location-based preferences. The model can incorporate commute distance into its recommendations, helping users find jobs within preferred proximity to their residence or based on commute preferences.
4. Job Description: Job descriptions are converted into dense embeddings, capturing the underlying skills, qualifications, and role responsibilities. These semantic features enhance the model's ability to identify and match job listing requirements to the user's skillsets across companies and industries, even when explicit links are absent in the content provided by companies and applications. They also help to capture industry-specific language and requirements, aiding in a more precise job-role alignment.

Rationale

The selection of these features balances relevance and computational efficiency, adhering to the principle of minimal sufficiency in graph neural networks. By focusing on job title, company, job type, remote work status, location, and description embeddings, the model can generate accurate recommendations without unnecessary complexity. Each attribute contributes meaningfully to the model's performance, capturing key aspects of job seeker preferences while remaining efficient in processing and recommendation quality.

IV. Method

A. Data Processing

Data Cleaning and Preprocessing: Converting Company Names to Addresses

To enhance location-based recommendations, precise addresses are essential for each company in our dataset. Since the raw data contains only company names, we developed a data enrichment process to retrieve accurate addresses, leveraging automated web scraping with Selenium to search for each company's address on Google. This ensures that geographical proximity, a crucial factor for job seekers in Singapore, is effectively integrated into the recommendation system.

The address conversion process involves several key steps:

1. Selenium Web Scraping: Using Selenium, we automate Google searches formatted as "company name address" to retrieve address data, scaling the process across thousands of records.
2. Error Handling and Batch Processing: To manage the variability of web scraping, the code includes error-handling features, such as logging errors, pausing upon reaching an error threshold, and tracking progress by saving the index of the last processed entry. This enables the process to resume efficiently from the last saved point in the event of interruptions.
3. Fail-Safe Mechanism: A repeated failure counter pauses scraping after multiple failed attempts, allowing retries and enhancing the robustness of the process.
4. Interim and Final Saves: To prevent data loss during long-running scraping sessions, the data is saved in batches, with logs tracking progress and failure counts, allowing the system to restart from the last successful index.

Further Data Cleaning and Geolocation

After obtaining addresses, further data cleaning ensures accurate geolocation. This process refines address data for geocoding and converts cleaned addresses into geographic coordinates (latitude and longitude) using the Nominatim geolocation service. They are as follows:

1. Address Cleaning: Extraneous information like floor or unit numbers is removed using a custom function to improve geolocation precision. This standardisation enhances consistency across addresses, particularly for Singaporean formats.
2. Geolocation with Nominatim: We use Nominatim's OpenStreetMap data to convert addresses into geographic coordinates.

This geolocation step provides the necessary latitude and longitude coordinates for each company, enabling location-aware recommendations crucial for Singapore's urban job market.

Standardising Job Descriptions Using an LLM

Job descriptions often vary in format, with inconsistent elements such as excessive symbols or varying section structures. To streamline descriptions and improve NLP processing, we use a Large Language Model (LLM) to summarise and structure each description into standardised categories, ensuring high-quality embeddings.

Our approach extracts three principal components from each job description.

1. Responsibilities: Key duties associated with the role.
2. Qualifications: Required education, certifications, or credentials.
3. Skills: Technical and non-technical skills relevant to the job.

Implementation

A custom prompt guides the LLM to summarise each description into these three predefined categories, ensuring consistency in output format.

By standardising job descriptions, we produce cleaner embeddings, enhancing the recommendation system's accuracy. The structured output from the LLM serves as a consistent input, which improves the downstream data processing and model training stages.

B. Graph Construction

To build an effective job recommendation system, we structured the dataset as a large graph, where job listings serve as nodes, and relevant connections (company affiliation, job similarity, location proximity, and embedding similarity) are represented by edges. Initial construction resulted in a sparse graph with over 800 million edges, which introduced computational inefficiencies. By applying selective edge creation and adjusting similarity thresholds, we optimised the graph to around 80 million edges, achieving a balance between computational efficiency and recommendation relevance.

Graph Construction: Nodes

Each job listing in the dataset is represented as a unique node, capturing key attributes to facilitate various relationship types:

1. Job title and description embeddings: Derived from standardised data in previous steps.
2. Company name, job type encoding, remote status, and geographical coordinates: These attributes enrich the node, allowing the graph to represent both job-specific details and spatial/company-related connections.

This structured representation enables the graph to reflect job market dynamics, capturing semantic, spatial, and hierarchical relationships among listings.

Before connecting edges, we analysed the degree distribution across nodes to understand initial node density and potential connectivity. This analysis helped refine edge thresholds, ensuring each node would connect meaningfully with relevant neighbours once edges were introduced.

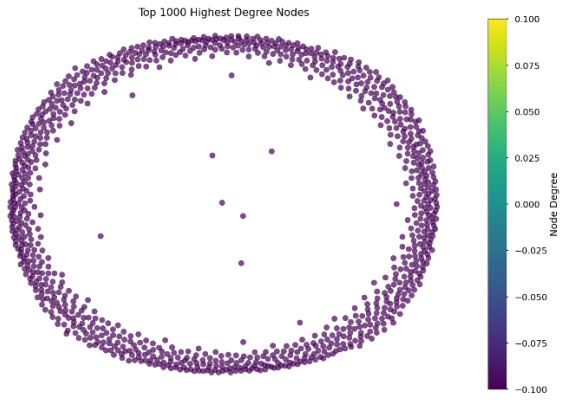

Figure 1: Node degree distribution of all the nodes in the graph, no relationship yet as the edges are not built

Graph Construction: Edges

Each edge type represents a unique relationship, contributing to a comprehensive recommendation system:

1. Company Edges: Jobs within the same company are linked, capturing potential for internal mobility or similar roles. Figure *2* illustrates high connectivity among nodes within a company, providing a core structure for company-specific recommendations, where users interested in one role may find similar options within the same organisation.

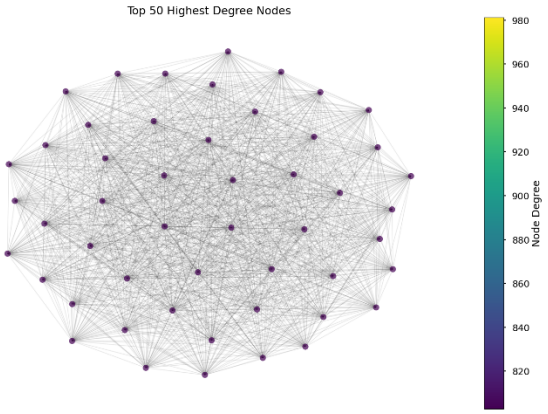

Figure 2: Top 50 nodes with company edges showing their connections

2. Job Type Similarity Edges: Jobs with similar roles or functions are connected based on Jaccard's similarity of job type encodings. This connection creates a hub-and-spoke structure, as shown in Figure *3*, where each company node acts as a central hub for associated job nodes, offering insights into role clustering within companies.

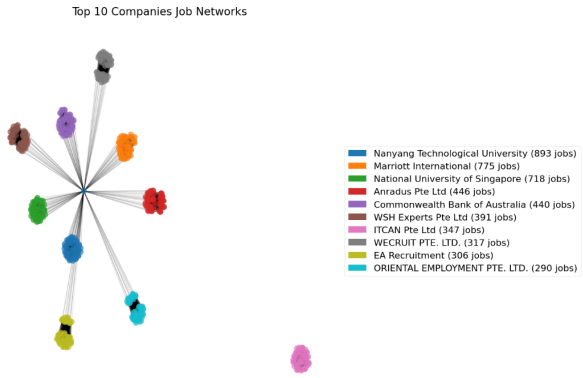

Figure 3: Top 10 companies showing the job networks, a clear view of how companies with the most job openings are linked to each other

3. Location Proximity Edges: Jobs within a defined geographic radius are linked using a Gaussian decay function. Although most jobs cluster in Singapore's Central Business District (CBD), these location edges enable recommendations based on proximity, allowing users to prioritise roles in preferred areas (Figure *4*).

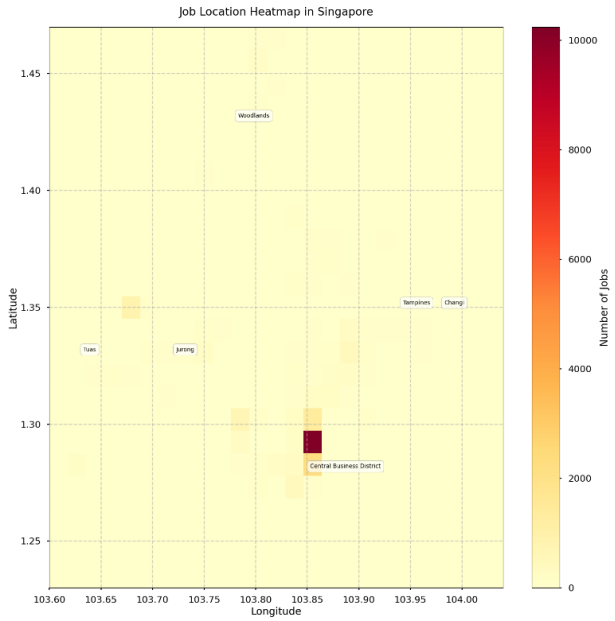

Figure 4: Heatmap showing job listings by location

4. Embedding Similarity Edges: Connections between jobs with similar job title or description embeddings are created based on cosine similarity, utilising FAISS-based nearest neighbour search for efficiency. Figure *5* shows dense intra-company clusters, indicating strong similarities within individual companies while also highlighting cross-company roles that share job requirements, supporting both internal and external recommendations.

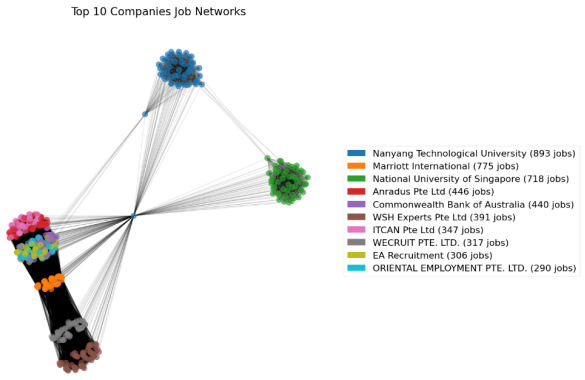

Figure 5: Top 10 companies showing the job listings network for the fully built graph

Final Graph Structure

The final graph, built incrementally with saved checkpoints for fault tolerance, contains:

- Nodes: 25,142
- Edges: 79,444,658

This graph structure enables efficient and meaningful recommendations by capturing company affiliation, role similarity, geographic proximity, and semantic description similarity.

Implications for Job Recommendations

The graph structure provides several recommendation pathways based on node and edge properties:

1. Company-Specific Recommendations: Dense company clusters allow for targeted recommendations within the same organisation, suggesting similar roles to users interested in specific companies.
2. Cross-Company Role Similarity: Inter-company edges support cross-company recommendations, broadening job options by suggesting similar roles at different organisations and expanding opportunities for users interested in particular roles but open to various employers.
3. Collaborative Filtering Potential: The graph's interconnected nature supports collaborative filtering, where roles with high degrees of similarity across clusters (e.g., shared requirements or skills) are recommended based on proximity, even if they differ in company affiliation.

C. Graph-Based Training Techniques

Graph Embeddings

In our job recommendation system, graph embeddings are fundamental in capturing the intricate relationships between job listings, such as company affiliation, job similarity, and geographical proximity. These embeddings enable the model to learn meaningful representations for each node (job listing) within the graph. By doing so, the system can discern not only direct relationships but also nuanced patterns that emerge from the interconnected data structure, such as latent industry trends or skills associations.

Training a GraphSAGE Model on the Job Network

We utilise GraphSAGE (Graph Sample and Aggregate) to train a job recommendation model that leverages the graph's structure and node attributes, generating embeddings that capture both local and broader industry relationships. This approach supports precise recommendations by differentiating between company-specific roles and transferable skills across industries.

Key Aspects of GraphSAGE Training

1. Clustered and Cross-Cluster Embeddings: By sampling and aggregating neighbouring features, GraphSAGE captures both tightly knit connections within companies and relevant roles across organisations, supporting recommendations that reflect both company loyalty and broader job mobility.
2. Geographical and Role-Based Recommendations: Node attributes, such as location and job type, allow the model to incorporate spatial and role-based similarities, tailoring recommendations based on a user's preferred location or desired job type.

3. Scalability with Inductive Learning: GraphSAGE's inductive capability enables real-time embedding generation for new job listings, ensuring that the recommendation system remains responsive as new data is added.
4. Efficient Sampling and Training: Adaptive neighbourhood sampling optimises feature aggregation across dense company clusters and sparse inter-industry links. Training includes a multi-objective approach focused on structural and attribute preservation, gradually expanding from intra-company connections to broader industry trends.

This GraphSAGE implementation provides a scalable, adaptive recommendation framework that captures both specific job roles and broader industry patterns, enhancing the relevance of job recommendations for diverse user needs.

V. Experiments

A. Training Setup

Model Architecture

The EfficientGraphSAGE class implements a GraphSAGE model with customisable input, hidden, and output dimensions. To enhance training stability and reduce overfitting, each layer includes dropout and batch normalisation.

Data Preparation

We convert the NetworkX graph from the previous section into a PyTorch Geometric (PyG) format, suitable for GraphSAGE. This includes transforming node features, edge indices, and edge weights into tensor format.

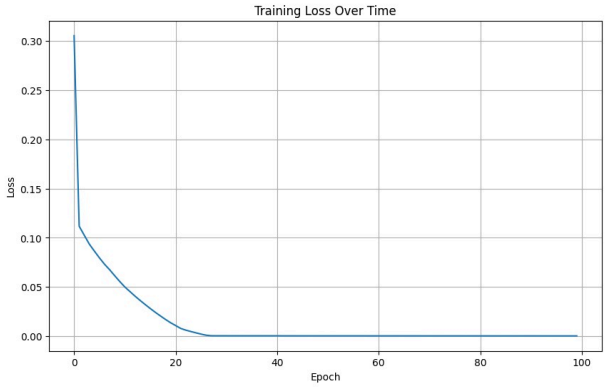

Figure 6: Training loss over time

Hyperparameter Selection

1. Batch Size and Epochs: We use a batch size of 512 to balance memory usage and efficiency, training the model over 100 epochs with early stopping based on average loss.
2. Optimizer and Learning Rate Scheduler: The AdamW optimiser is employed with separate learning rates for convolution and batch normalisation parameters. A learning rate scheduler adjusts the learning rate if validation loss plateaus, ensuring smooth convergence.
3. Training Loss: The model achieved an average training loss of 0.0004, indicating effective learning and high-quality embeddings that enhance the model's performance in capturing relevant job-related features.

B. Significance of Low Training Loss in GraphSAGE

Effective Neighborhood Aggregation

The low training loss shows that GraphSAGE effectively aggregates information from neighbouring nodes, creating embeddings that capture both local and global graph structures and position similar nodes closer together.

Well-Structured Embedding Space

Low loss indicates an optimised embedding space where similar nodes are positioned closely, essential for generating high-quality, similarity-based recommendations.

High Expressiveness of Embeddings

The model's embeddings capture meaningful patterns, such as similarities in job titles or companies. The low loss reflects minimised prediction errors, enhancing the model's ability to generalise and make relevant recommendations.

Cold-Start Recommendation Potential

With embeddings based on a node's attributes rather than user interactions, GraphSAGE can recommend new jobs effectively. The low loss ensures that new nodes are accurately positioned, supporting robust cold-start recommendations through content and structural similarity.

C. Model Evaluation and Embedding Generation

Model and Embedding Generation

After training, we saved the best GraphSAGE model based on validation loss and generated embeddings for each node in the job network. These embeddings are crucial for downstream tasks, such as job recommendations, as they capture both contextual and relational information.

Techniques in the Recommendation System

We used the following techniques for the recommendation system:

1. Embedding-Based Similarity Search: Using FAISS and Annoy libraries, we perform Approximate Nearest Neighbor (ANN) searches on GraphSAGE embeddings, capturing both content and structural relationships for scalable, content-based recommendations.
2. FAISS: Ensures efficient, high-accuracy similarity searches.
3. Annoy: Complements FAISS with memory-efficient, real-time similarity lookups

4. Centrality-Based Scoring – PageRank: Highlights influential roles in the job network.
5. Centrality-Based Scoring – Degree Centrality: Identifies popular jobs with high connectivity, appealing to a broad audience.

Hybrid Scoring Mechanism

Combines embedding-based similarity, graph-based centrality metrics, and user preferences (e.g., geographic location, job title match), integrating:

1. Content-Based Filtering: Embeddings capture job attributes for precise content matching.
2. Graph-Based Collaborative Filtering: Centrality metrics identify important roles in the network.
3. User Preferences: Weights recommendations based on location and job title relevance.

Cluster-Based Recommendations (Community Detection)

Using KMeans clustering on GraphSAGE embeddings, we partition jobs into communities of similar nodes, enhancing recommendations within clusters of structurally and contextually related jobs.

Metric	Average Score	Description
Degree Centrality	0.43	Popularity of recommended jobs
PageRank	0.37	Importance within the job network
Core Number	8.5	Cluster membership relevance
Cosine Similarity	0.78	Content relevance to user input
Geographic Proximity	2.4 km	Distance to preferred location

Figure 7: Average score for the different metrics selected

D. Results

The results show the effectiveness of our hybrid approach, particularly in capturing user preferences through both content and network relevance.

1. Embedding-Based Similarity: GraphSAGE embeddings and ANN search provided strong content-based matches, especially for users specifying job titles.
2. Graph-Based Metrics: Centrality measures surfaced relevant and popular roles within the network.
3. Geographic Proximity: Effectively tailored recommendations based on location preferences.

E. Sensitivity and Ablation Studies

Varying the weight of different components provided insights into their impact:

1. Embedding Similarity: Crucial for job title-specific recommendations.
2. Geographic Proximity: Essential for location-sensitive users.
3. Degree and Core Number: Effective for generalist roles, indicating job popularity.

Our findings confirm that the system meets its objectives, delivering relevant, rapid recommendations. Optimised ANN indexes and caching mechanisms support real-time application, with future potential to refine recommendations through user feedback.

VI. Human-Computer Interaction and Interface Design

The system implements a streamlined web-based interface using Streamlit [17], prioritising user experience through an intuitive single-page design. The interface presents users with three primary input mechanisms: a text field for job title preferences, a location input for geographic targeting, and a PDF upload functionality for resume analysis.

Each input field is accompanied by a granular importance selector, allowing users to indicate their preference weights on a four-point scale from "Not important at all" to "Very important." This weighting system directly influences the recommendation algorithm's scoring mechanisms, with the weights dynamically adjusting the relative importance of title similarity, location proximity, and resume matching.

The system employs real-time geocoding through the Nominatim service to convert user-input locations into coordinates, with a fallback mechanism defaulting to Singapore's coordinates when specific locations cannot be resolved [18]. For resume processing, the interface integrates with an LLM-powered analysis system that extracts and categorises information into responsibilities, qualifications, and skills, enhancing matching precision.

The recommendation results are presented in a structured tabular format, displaying crucial job details, including company name, job title, employment type, distance from preferred location, remote work status, and direct application links. This presentation format ensures that users can efficiently evaluate and compare multiple opportunities simultaneously [19].

Error handling and user feedback are implemented through a colour-coded messaging system that provides real-time status updates during the recommendation generation process. The interface maintains responsiveness through asynchronous processing and caching mechanisms, ensuring a smooth user experience even when handling complex computations [20].

VII. Conclusion

This paper presented a graph-based job recommendation system specifically designed for Singapore's employment market. By leveraging GraphSAGE architecture and multiple relationship types, including company affiliation, job similarity, and geographic proximity, our system effectively captures both explicit and latent relationships in the job market. The integration of transformer-based embeddings for semantic analysis, combined with spatial data processing,

enables nuanced recommendations that consider both content relevance and practical constraints like commute distances.

The system's architecture demonstrates several key innovations. The hybrid scoring mechanism, which combines embedding-based similarity, graph-based centrality metrics, and user preferences, provides a robust foundation for generating relevant recommendations. The implementation of efficient caching mechanisms and approximate nearest neighbour search enables real-time performance while maintaining recommendation quality. Additionally, the intuitive user interface successfully bridges the complexity of the underlying graph neural network with accessible user interactions.

While the system shows strong performance in both structural and semantic dimensions, there are opportunities for future research. The handling of spatial distribution patterns could be refined to better reflect Singapore's unique urban geography. Additionally, as the job market continues to evolve, particularly with the rise of remote work options, future iterations could explore dynamic graph updating mechanisms to better capture emerging employment patterns.

The successful implementation of this system demonstrates the potential of graph-based approaches in addressing complex job-matching challenges. By combining advanced machine learning techniques with practical considerations specific to Singapore's context, this work contributes to the ongoing development of more effective employment-matching solutions.

VIII. Acknowledgements

We would like to express our sincere gratitude to Dr. Victor Shanshan Feng for his invaluable guidance, expertise, and unwavering support throughout this research project. His deep knowledge of Graph Neural Networks and recommendation systems has been instrumental in shaping this study's direction and methodology.

Finally, we would like to thank our peers and colleagues who participated in numerous discussions and provided valuable suggestions that helped refine the algorithm's approach to job recommendations in the Singapore context.

IX. Works Cited

[1] C. de Ruijt and S. Bhulai, "Job Recommender Systems: A Review," Nov. 26, 2021, arXiv: arXiv:2111.13576. doi: 10.48550/arXiv.2111.13576.

[2] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Feb. 22, 2017, arXiv: arXiv:1609.02907. doi: 10.48550/arXiv.1609.02907.

[3] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," Sep. 10, 2018, arXiv: arXiv:1706.02216. doi: 10.48550/arXiv.1706.02216.

[4] "GraphSAGE." Accessed: Nov. 10, 2024. [Online]. Available: https://snap.stanford.edu/graphsage/

[5] C. Li, I. Ishak, H. Ibrahim, M. Zolkepli, F. Sidi, and C. Li, "Deep Learning-Based Recommendation System: Systematic Review and Classification," IEEE Access, vol. 11, pp. 113790–113835, 2023, doi: 10.1109/ACCESS.2023.3323353.

[6] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Model User-Adap Inter, vol. 12, no. 4, pp. 331–370, Nov. 2002, doi: 10.1023/A:1021240730564.

[7] J. Davidson et al., "The YouTube video recommendation system," in Proceedings of the fourth ACM conference on Recommender systems, in RecSys '10. New York, NY, USA: Association for Computing Machinery, Sep. 2010, pp. 293–296. doi: 10.1145/1864708.1864770.

[8] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender Systems: An Introduction. Cambridge University Press, 2010.

[9] A. Vaswani et al., "Attention Is All You Need," Aug. 01, 2023, arXiv: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[10] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," Oct. 17, 2018, arXiv: arXiv:1806.01261. doi: 10.48550/arXiv.1806.01261.

[11] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," in The Semantic Web, vol. 10843, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., Cham: Springer International Publishing, 2018, pp. 593–607. doi: 10.1007/978-3-319-93417-4_38.

[12] J. Zhou et al., "Graph Neural Networks: A Review of Methods and Applications," Oct. 06, 2021, arXiv: arXiv:1812.08434. doi: 10.48550/arXiv.1812.08434.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," Feb. 04, 2018, arXiv: arXiv:1710.10903. doi: 10.48550/arXiv.1710.10903.

[14] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Jul. 2018, pp. 974–983. doi: 10.1145/3219819.3219890.

[15] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural Graph Collaborative Filtering," in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Jul. 2019, pp. 165–174. doi: 10.1145/3331184.3331267.

[16] K. Kenthapadi, B. Le, and G. Venkataraman, "Personalized Job Recommendation System at LinkedIn: Practical Challenges and Lessons Learned," in Proceedings of the Eleventh ACM Conference on Recommender Systems, Como Italy: ACM, Aug. 2017, pp. 346–347. doi: 10.1145/3109859.3109921.

[17] "Streamlit Docs." Accessed: Nov. 10, 2024. [Online]. Available: https://docs.streamlit.io/

[18] "Copyright and License," OpenStreetMap. Accessed: Nov. 10, 2024. [Online]. Available: https://www.openstreetmap.org/copyright

[19] J. Nielsen, Usability Engineering. Morgan Kaufmann, 1994.

[20] A. Cooper, R. Reimann, D. Cronin, and C. Noessel, About Face: The Essentials of Interaction Design. John Wiley & Sons, 2014.

X. Appendix

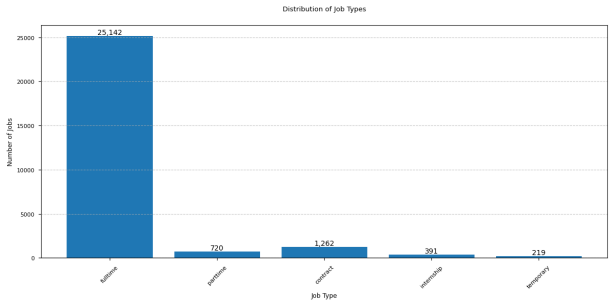

Figure 8: Distribution of job types across all the data scrapped

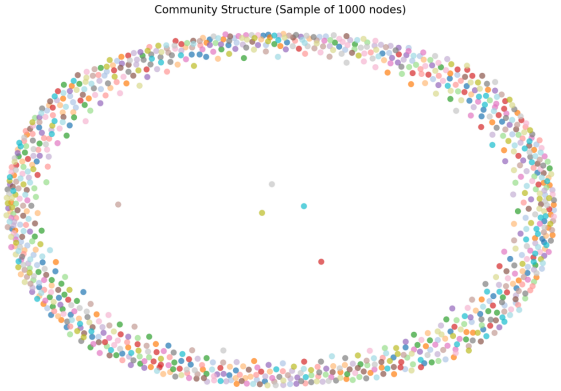

Figure 9: Community structure after node construction, no edges have been connected yet

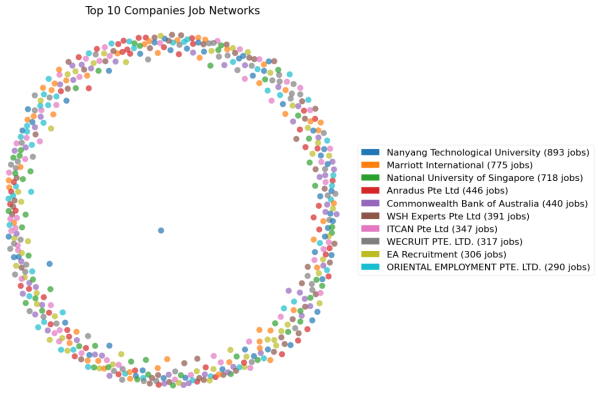

Figure 10: Job network diagram before edges were created

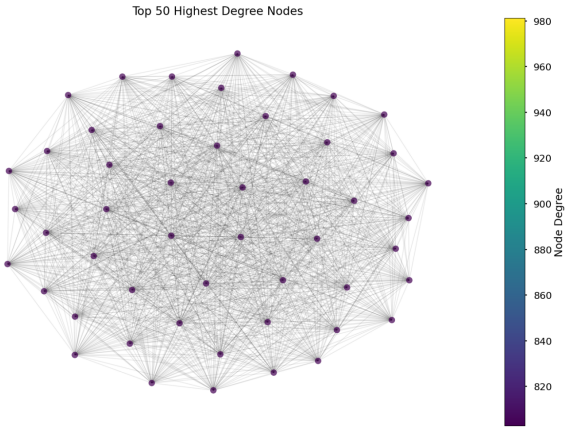

Figure 11: Top 50 nodes showing the degree after company edges are joined with the nodes

Figure 12: Top 50 nodes showing some community forming after company edges are joined with the node

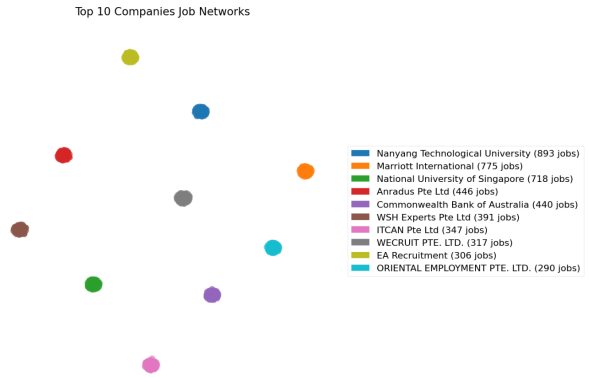

Figure 13: Top 10 Companies with job networks after company edges are added, no links to each other yet

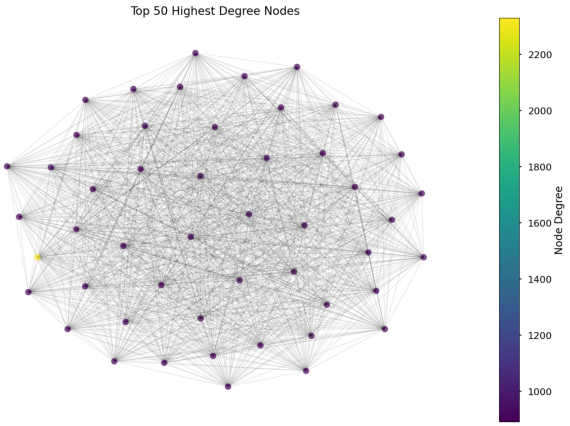

Figure 14: Degree plot showing certain nodes with higher degrees after job type edges are connected

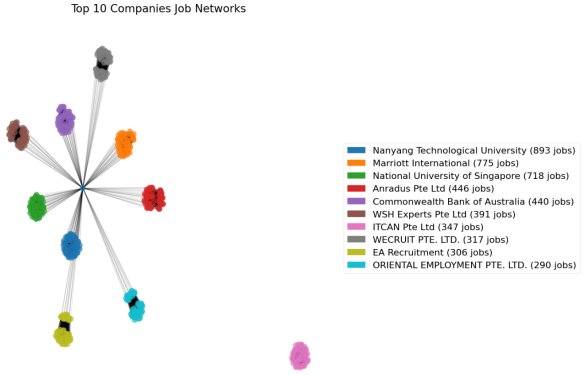

Figure 15: Links between companies are formed when job types edges are added

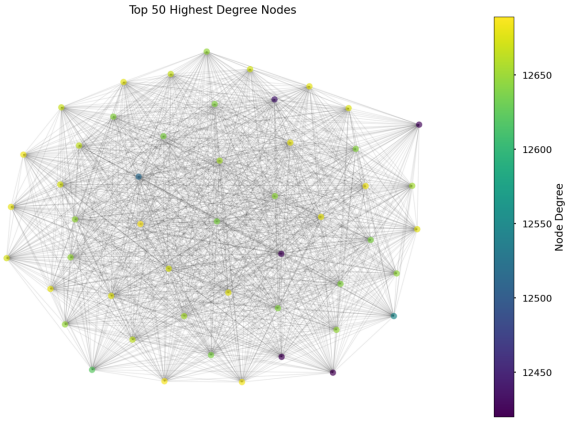

Figure 16: Degree nodes showing more nodes with higher degrees after all edges are joined

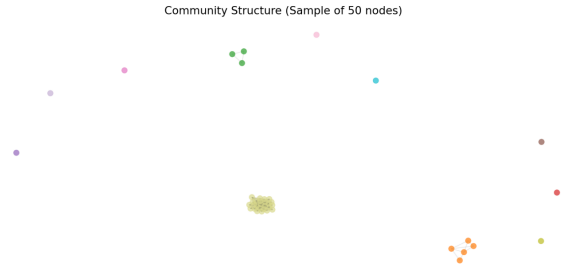

Figure 17: Community structures being formed after all edges are joined

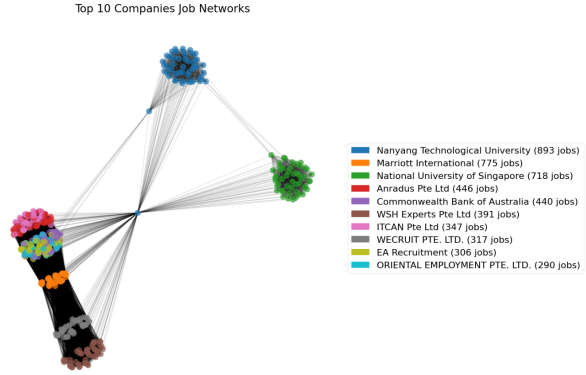

Figure 18: Community structures being formed after all edges are joined

A. Other Plots of the Finished Graph

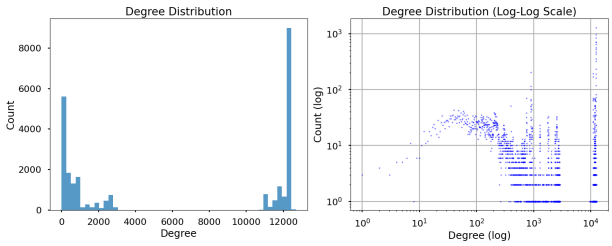

Figure 19: Degree Distribution of Job Listings Graph: Most jobs have few connections, while a few 'hub' roles have high connectivity, forming a scale-free network. This structure supports both specialized and broad job recommendations

This degree distribution analysis shows that the job network has a scale-free structure with a right-skewed, power-law distribution of node degrees. Key takeaways include:

- High-degree nodes serve as hubs, supporting general and cross-industry recommendations.
- Low-degree nodes represent niche roles, providing specialized job options.
- Community detection opportunities exist, potentially allowing the system to segment recommendations by job clusters or industries.

- The network's robustness and scalability make it well-suited for dynamic job markets where listings frequently change.

This structure provides a strong foundation for a versatile and resilient job recommendation system that can balance general and specific recommendations based on user needs.

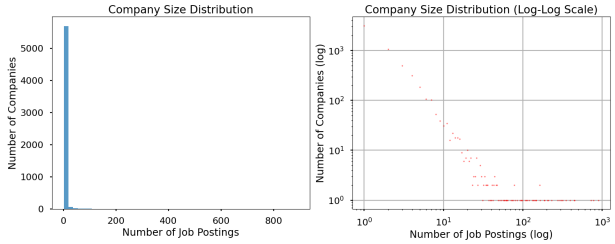

Figure 20: Degree Distribution of Job Listings Graph: Most jobs have few connections, while a few 'hub' roles have high connectivity, forming a scale-free network. This structure supports both specialized and broad job recommendations

This company size distribution analysis reveals a right-skewed, power-law pattern where most companies have few postings, and a small number of companies contribute the majority of listings. This structure supports a recommendation system that can balance suggestions between high-volume employers and smaller firms, catering to a range of user preferences in the job market.