THOMAS Romain

<u>Promotion « 2025 »</u>

Master of Science in
Data Science & Artificial Intelligence Strategy

Enhancing Session-Based Job Recommender Systems with Autoencoders and Large Language Models

Date of Defense: June 2025

Name of Supervisor: Saeed VARASTEH YAZDI

Abstract

In an increasingly anonymous and dynamic job search environment, session-based recommender systems (SBRS) have become essential for capturing short-term user intent. Building upon the foundational work by Lacic et al. (2020), this thesis proposes an enhanced SBRS for job recommendation that leverages semantic embeddings and multimodal data fusion. Lacic et al.'s study demonstrated that autoencoders, particularly variational autoencoders, effectively generate latent session representations for anonymous job recommendation by combining session interactions with one-hot encoded topic features extracted from job descriptions. In contrast to the use of one-hot encoded topic models for job content, this study replaces these representations with contextual embeddings derived from Bidirectional Encoder Representations from Transformers (BERT), enabling richer semantic modeling of job descriptions, titles, and requirements.

To evaluate the impact of this enhancement, classical, denoising, and variational autoencoder (VAE) architectures were trained on session vectors composed of both BERT embeddings and structured metadata, using the CareerBuilder 2012 dataset. Session representations were constructed by concatenating embedded job vectors within a session, yielding fixed-size inputs for training. The encoded latent vectors were then used with a k-nearest neighbor approach to generate recommendations.

Model performance was assessed using traditional accuracy metrics (Normalized Discounted Cumulative Gain (nDCG), Mean Reciprocal Rank(MRR)) and beyond-accuracy metrics (Expected Popularity Complement, Expected Profile Diversity, item coverage, and session coverage). Compared to Lacic et al.'s results, BERT-enhanced models significantly improved recommendation accuracy across all autoencoder variants. However, improvements in novelty and diversity were limited, suggesting that semantic richness alone is insufficient to drive beyond-accuracy gains without additional diversity-oriented constraints or re-ranking mechanisms.

These findings highlight the importance of integrating large language models into recommender pipelines while acknowledging the continued challenge of optimizing for novelty and diversity in session-based recommendations. The thesis demonstrates that a VAE combined with BERT embeddings offers a robust and scalable framework for job recommendation, particularly when accuracy and semantic alignment are critical.

Table of content

List of figures

List of tables

Chapter 1 - Introduction

In an increasingly dynamic and competitive job market, online job portals and professional social networks have emerged as critical intermediaries connecting job seekers with employment opportunities. However, matching candidates to relevant job listings remains a complex challenge due to limited user data, evolving job descriptions, and the need for recommendations that are not only accurate but also novel and diverse. This challenge is particularly pronounced in session-based environments, where users browse anonymously and little to no historical data is available to personalize recommendations.

Traditional recommender systems often rely on collaborative filtering or content-based methods. While effective in domains with rich interaction histories, these approaches falter when applied to transient, anonymous user behavior typically observed in job search scenarios. To address this gap, recent research has turned to deep learning models capable of capturing latent user intent from short-lived sessions. Among these, autoencoders have shown promise by learning compressed representations of session activity, which can be leveraged for recommendation through similarity-based retrieval.

This thesis builds upon the session-based job recommendation framework introduced by Lacic et al. (2020), which demonstrated that autoencoders, particularly variational autoencoders (VAEs), can provide strong performance in terms of accuracy, novelty, and coverage. While their model combined session interactions with binary-encoded job content derived from topic modeling, it did not leverage the full semantic richness embedded in job descriptions. Consequently, the recommendations may miss contextual cues essential for aligning user preferences with job attributes.

To overcome these limitations, this study proposes an enhanced session-based recommender system that incorporates semantic embeddings derived from BERT, a state-of-the-art language model. By replacing one-hot topic vectors with dense BERT-based representations of job descriptions, titles, and requirements, we aim to capture deeper semantic relationships between jobs and user sessions. Additionally, the system integrates structured job features, enabling a multimodal data fusion strategy that enriches the session representation.

Through a series of experiments on the CareerBuilder 2012 dataset, this thesis evaluates the performance of classical, denoising, and variational autoencoder architectures trained on BERT-augmented session vectors. Using both accuracy (nDCG, MRR) and beyond-accuracy metrics (EPC, EPD, coverage), we compare the enhanced models against benchmarks established in prior work. The results provide insights into the benefits and limitations of integrating large language models and multimodal data into session-based recommender systems.

In doing so, this research contributes to the growing field of intelligent e-recruitment systems by exploring how semantic and multimodal enhancements can improve recommendation quality in anonymous, data-sparse environments. Ultimately, the findings support the development of fairer, more diverse, and contextually aware job recommendation platforms.

Contributions and Findings

- Replace one-hot topic vectors with contextual BERT-based embeddings for job titles, descriptions, and requirements, enabling richer semantic modeling.
- Implement and compare classical, denoising, and variational autoencoders trained on these embeddings, assessing their suitability for session-based recommendation.
- Demonstrate how combining structured metadata (e.g., job state) with semantic embeddings improves robustness and coverage in a multimodal learning setting.
- Evaluate the models on the CareerBuilder 2012 dataset using both traditional accuracy metrics (nDCG, MRR) and beyond-accuracy metrics (EPC, EPD, item coverage, session coverage).
- Show that BERT-enhanced models significantly outperform topic-based baselines in accuracy and coverage, while VAE-based models remain competitive in novelty and robustness.

Organization of the Paper

- Part 1 - Literature Review** explores the evolution of job recommender systems, session-based methods, autoencoder architectures, large language models, and multimodal fusion techniques.
- Part 2 - Methodology** details the research design, data preprocessing, model architecture, and recommendation logic, including enhancements over the baseline.
- Part 3 - Experiments and Evaluation** presents the setup, dataset characteristics, and evaluation metrics used to benchmark model performance.
- Part 4 - Results and Discussion** compares the proposed models to Lacic et al. (2020), analyzes outcomes, and revisits the hypotheses.

Chapter 2 - Literature review

Content:

2.1 Job Recommender Systems

Job recommender systems (JRS) are algorithmic tools designed to bridge the gap between job seekers and employment opportunities. By analyzing user preferences, qualifications, and behaviors, these systems propose personalized job matches that streamline the hiring process and enhance the experience of job discovery (Lundberg et al., 2021).

2.1.1 Functional Overview and Approaches

The foundational models used in job recommender systems mirror those in general recommendation literature.

Collaborative Filtering (CF) leverages the historical behaviors of users to find patterns and make job recommendations based on the preferences of similar users (Bobadilla et al., 2013). Typically operationalized via user-item interaction matrices, CF methods can be further classified into memory-based and model-based approaches. The former relies on KNN-style similarity, while the latter involves latent factor models or embeddings learned from interaction data (de Ruijt & Bhulai, 2021; Al-Otaibi & Ykhlef, 2012).

Content-Based Filtering (CBF)relies on structured and unstructured job-related data (e.g., skills, titles, industries) to recommend jobs aligned with a user's profile (Pazzani & Billsus, 2007). A key challenge in content-based JRS is vocabulary mismatch. Job seekers and recruiters may describe skills or roles differently, leading to poor semantic alignment (de Ruijt & Bhulai, 2021).

Hybrid methods approaches combine collaborative filtering and content-based filtering to leverage the strengths of both and mitigate their individual weaknesses (Burke, 2002). Hybrid systems can be implemented in two main ways:

- Monolithic hybrids merge CF and CBF at the algorithmic level, often using a unified model such as a neural network or matrix factorization approach that jointly learns from both interaction and content data. This enables deep integration and end-to-end optimization but may reduce flexibility and interpretability.
- Ensemble hybrids, by contrast, keep CF and CBF models separate and combine their outputs using strategies like score weighting, rule-based switching, or meta-level stacking (Aggarwal, 2016). These systems are more modular and interpretable, and allow easy integration of additional signals such as ontology-based features or probabilistic models (Al-Otaibi & Ykhlef, 2012).

Ensemble hybrids, combined with transfer learning approaches, are particularly advantageous for cold-start scenarios and building multimodal extensions, such as connecting user interaction data with embeddings derived from NLP processes.

Over the last several years, job recommender systems have turned increasingly to deep learning- and knowledge-based systems, with deep learning systems that better connect resumes to job descriptions semantically (BERT, Siamese Networks) providing improved relevance and accuracy (de Ruijt & Bhulai, 2021). An important evolution in our thinking is the move to a challenge-oriented approach to system design, where design choices are guided by specific challenges (cold start, fairness, reciprocity, etc.). As highlighted by Mashayekhi et al. (2023), we are beginning to develop a field that is not just concerned with types of models, but on building technical design strategies that best address issues faced by real-world recommendation systems.

Closely aligned, Lacic et al. (2020) developed a session-based recommendation system using autoencoders. They encode short, anonymous sessions of user activity into latent vectors, and their variational autoencoders perform well in terms of traditional accuracy measures as well as alternative measures like novelty and coverage. While they provide a baseline for comparison, their model only utilized implicit user activity data, and, could have used job content information to enhance their model. In this thesis, I would like to expand their work and investigate whether merging some NLP-based information into the model can improve upon their findings, especially related to semantic and contextual facets of job postings.

2.1.2 Design Taxonomies: Informing Architecture through Problem Framing

Taxonomies in job recommender systems serve as analytical frameworks that guide system design and clarify how different approaches address the unique challenges of the job domain. Rather than simply categorizing techniques, these taxonomies help researchers and practitioners reason about the suitability and limitations of various methods in specific scenarios.

For instance, understanding whether a system is designed around collaborative filtering, content-based filtering, or hybrid techniques determines its ability to handle sparsity and cold start problems. Cold start situations are particularly common in job recommendation. They are better addressed through content-based or ontology-enhanced models, which do not rely on historical interaction data.

The axis of embedding methods is also crucial. Shallow embeddings like TF-IDF or word2vec may suffice for capturing keyword-level semantics. But deep embeddings, such as those learned via autoencoders or transformers, enable richer semantic understanding and generalization particularly when user history is absent or sparse. This directly informs the decision to base the present thesis on variational autoencoders, with planned enhancements via natural language processing (NLP) to more effectively utilize job description text.

User interaction types (explicit vs. implicit) and session-awareness are particularly relevant for this work. In job portals where many users browse anonymously, session-based systems are essential. Traditional user-based CF techniques fail in such cases, reinforcing the need for latent session modeling methods, as demonstrated by Lacic et al. (2020).

Additionally, the system design axis (monolithic vs. ensemble) reflects operational complexity and modularity. Ensemble systems, while more complex, may offer superior flexibility in integrating multimodal features. This feature that will be leveraged in this thesis to combine job metadata with text-based embeddings from NLP models.

Finally, directionality intersects directly with fairness and reciprocity concerns. Bidirectional systems, which consider both job seeker and recruiter preferences, align more closely with equitable matching goals. This highlights potential future extensions of work.

In summary, taxonomies not only provide structural understanding but also inform design choices and methodological priorities. In this thesis, the chosen configuration is: deep learning-based, session-aware, content-augmented, and challenge-driven. It is directly justified by this taxonomical reasoning.

2.1.3 Key Challenges in Job Recommender Systems

Several challenges uniquely affect the domain of job recommendation:

- Reciprocity: Unlike standard recommender systems, job platforms must account for mutual interest. Jobs must appeal to candidates, and candidates must be suitable for employers. This dual-sided nature requires models that balance preferences and suitability (Mashayekhi et al., 2023).
- Temporality: Job availability and user preferences are highly dynamic. Recommender systems must be temporally aware, adapting quickly to changes in job listings and candidate behavior. Features such as posting age, session history, or predicted job-switch timing can significantly improve relevance (de Ruijt & Bhulai, 2021).
- Fairness: Biases in job recommenders can perpetuate gender or racial inequalities. Simply removing sensitive features is insufficient; fair algorithms must actively audit and mitigate disparate outcomes (Chen et al., 2018; Geyik et al., 2019). Bias can appear in interaction patterns, exposure distributions, or ranking processes, and may need two-sided fairness models to address both job seekers and employers (Mashayekhi et al., 2023).
- Labor Market Impact: Recommender systems shape who sees what job, thus influencing market dynamics. Over-recommending certain jobs may cause applicant congestion and skew hiring trends, potentially harming diversity and fairness (Borisyuk et al., 2017).
- Cold Start and Sparsity: Job domains suffer from high sparsity due to the unique and evolving nature of both candidate and job profiles. Content-based features, clustering, and graph densification are promising approaches to reduce cold start impact (Mashayekhi et al., 2023).
- Multi-stakeholder Objectives: Unlike traditional systems, JRS must balance multiple stakeholders: job seekers, recruiters and platforms. This requires designing algorithms that consider bidirectional matching success and optimizing for fair exposure and recruitment likelihood (Mashayekhi et al., 2023).
- Session-based Recommendation: Many users interact with job platforms anonymously, which restricts user profiling. Lacic et al. (2020) tackle this by modeling sessions using autoencoder embeddings. However, the system does not integrate textual job descriptions through NLP models.

2.1.4 Ethical and Practical Considerations

Besides technical performance, ethical considerations are paramount:

- Bias Mitigation: Techniques like two-sided fairness (Sonboli et al., 2021) and demographic parity adjustments are needed to ensure equitable exposure across candidate groups.
- Privacy Preservation: Techniques like differential privacy and federated learning are increasingly explored to protect user data, especially as interaction logs and resumes can contain sensitive personal information (McMahan et al., 2017).
- Validation Practices: Due to data scarcity, many JRS are validated using competition datasets (e.g., CareerBuilder 2012, RecSys 2016/2017). However, model performance varies significantly across datasets, raising concerns about generalizability (Lacic et al., 2020).
- Transparency and User Control: Enhancing explainability and allowing users to understand or customize their recommendations fosters trust. This is especially important in high-stakes contexts like employment.

By addressing these challenges and incorporating ethical principles, future job recommender systems can become not only more effective but also more equitable and trustworthy.

2.2 Recommendation Systems Based on Sessions

2.2.1 Overview: Variations in Emergence and Classic Recommendation Techniques

Long-term user profiles and historical interaction data are mostly relied upon in traditional recommender systems including content-based approaches and collaborative filtering. But in many real-world situations, including first-time users on streaming platforms or anonymous users browsing e-commerce sites, this historical data is either completely lacking or very rare. SBRS emerged to solve this restriction, emphasizing on the user's behavior inside a single interaction session instead of long-term data (Hidasi et al., 2016).

Often buried in recent interactions inside a session rather than long-term historical data, session-based recommender systems (SBRSs) seek to capture short-term and dynamic user preferences. A session, according to Wang et al. (2021), is a limited series of user interactions that co-occur inside a given context and time, say a product browsing session or a music streaming session. SBRSs are perfect for anonymous and cold-start situations since unlike conventional recommender systems they do not assume the availability of user profiles or IDs.

Because they are relevant in practical settings where the main goal is to forecast the next interaction in an ongoing session, SBRSs have become rather well-known. By means of a taxonomy of SBRS techniques and challenges, Wang et al. (2021) highlight the variety of session data traits including session length, order sensitivity, and context-dependence. These complexity call for specific models able to represent intra-session dependencies independent of long-term behavioral history. Jannach (2018) underlined even more that one of the main drivers behind SBRS research is the necessity to modify recommendation strategies to short-term user intent, which could vary greatly from past behavior and must usually be deduced from few data. For use cases like e-commerce, where users show goal-driven behavior inside a single session, this short-term intent is especially important since contextual signals like recency or sequence patterns become more useful than global preferences.

To forecast the next most likely item of interest, SBRS examine the user action sequence during a session (clicks, views, scrolls). This method aligns well with present privacy and data minimization trends by allowing real-time, tailored recommendations without depending on constant user identities (Wang et al., 2021).

2.2.2 Session-Based Method Most Popular

Wang et al. (2021) classify in their thorough survey into several families: conventional methods (KNN, Markov Chains), latent representation learning methods (Matrix Factorization, Autoencoders), sequence modeling (RNNs, CNNs), and graph-based methods (GNNs). They also point up hybrid techniques combining several paradigms.

Their work highlights the need of extracting the contextual signals as well as the sequential dynamics found in session data. For instance, whereas sequential models such as RNNs and GNNs gain from attention mechanisms to capture user intent, latent models can be improved by including side information. The paper pins open research difficulties including modeling unordered sessions, adjusting to changing intent, and reaching explainability.

Deep learning, especially models meant to process graphs and sequences, has driven SBRS's evolution. These models capture the sequential or structural character of user behavior in sessions, so often surpassing conventional baselines (Jannach, 2018).

2.2.2.1 KNN-Based Approach:

Conventional yet strong for session similarity modeling. Using cosine or Jaccard distance over item co-occurrence, KNN approximates the similarity between the current session and past sessions. Though basic, these techniques are quite interpretable and effective; they usually act as competitive baselines in SBRS benchmarks (Ludewig & Jannach, 2018).

2.2.2.2 RNN-Based approaches

Especially GRU4Rec (Hidasi et al., 2016), recurrent neural networks (RNNs) reached a significant turning point by including Gated Recurrent Units (GRUs) to replicate user session sequences. RNNs adapt to changing user behavior inside a session by capturing temporal dependencies rather well. Apart from GRUs, Long Short-Term Memory (LSTMs) have also been extensively applied for session-based recommendation tasks. By reducing the vanishing gradient problem via gated structures controlling information flow, LSTMs can learn long-range dependencies. LSTMs assist model not only immediate past actions but also more distant interactions inside a session, so offering a richer knowledge of sequential patterns when sessions are longer and user behaviors are more complicated (Wang et al., 2021).

Incorporating ranking loss functions such as Bayesian Personalized Ranking (BPR) and TOP1 to rank pertinent items, the GRU4Rec model was tailored for recommendation tasks. Hidasi et al. also suggested negative sampling methods and session-parallel mini-batches to effectively manage varying session lengths

and big item spaces.
On public datasets, this approach showed notable gains over more traditional approaches and motivated a broad spectrum of extensions and hybrid models including contextual inputs and attention mechanisms (Wang et al., 2021).

2.2.2.3 GNN-Based Strategies

For simulating the session as a graph of user-item interactions. Graph neural networks (GNNs) model sessions as graphs where items are nodes and transitions are edges. Techniques such as Session-based Recommendation with Graph Neural Networks (SR-GNN) (Wu et al., 2019) treat each session as a directed graph in which nodes match objects and edges reflect changes between successive item clicks. SR-GNN learns item embeddings by aggregating data from adjacent nodes using a Gated Graph Neural Network (GGNN), so capturing both local and global dependencies inside the session.
SR-GNN fused using an attention mechanism two elements of user behavior: the user's most recent interest and their general inclination over the session. Especially suited for anonymous session-based environments, this method avoids depending on user profiles or persistent identifiers. Extensive studies on benchmark datasets showed that in accuracy and ranking performance SR-GNN greatly beats conventional RNN and Convolutional Neural Network (CNN) based models.
When handling complex browsing patterns and sparse datasets where linear assumptions or fixed sequential orders are inadequate to capture user behavior, GNN-based models are especially strong.

2.2.2.4 Based on Autoencoder

Learning latent representations helps autoencoder-based models to recreate user sessions. These models decode session information from a compact vector to forecast next objects. Variational Autoencoders (VAE) and Denoising Autoencoders (DAE), which increase resilience to noise and incomplete session data, are among variants.
Using autoencoders for job recommendations in session-based settings where user history is either limited or absent was investigated by Lacic et al. (2020). Their method encoded sessions into latent vectors using three distinct autoencoder architectures. After that, job recommendations were based on k-nearest neighbor models of these representations. Autoencoders can outperform other session-based techniques not only in accuracy but also in beyond-accuracy measures, evaluations on datasets such as Career Builder and XING revealed. Furthermore, the autoencoder input's inclusion of job posting materials let for a fusion of interaction and content signals, so improving recommendation performance. Autoencoders are quite appropriate for job recommendation situations and complement this thesis's interest in integrating multimodal data since they allow one to combine several modalities.
Hybridizing with big language models helps autoencoders which are especially helpful for integrating multimodal signals like textual descriptions, timestamps,...

2.2.2.5 STAMP-Based Approachologies

The STAMP model is an acronym for Short-Term Attention/Memory Priority and was developed by Liu et al., 2018. STAMP combines a memory mechanism with an attention layer. It was created to record the overall interest of the users. Unlike RNNs or GNNs, STAMP takes advantage of a simpler Multi-Layer Perceptron (MLP) architecture improved by attention enabling it to learn from the most relevant interactions.
For each session, STAMP produces two representations: a general interest vector obtained from the session context; and a short-term interest vector generated from the last clicked item. An attention mechanism then merges the two vectors by dynamically ranking user behavior, either long-term or short-term. This framework allows STAMP to generate improved next-click predictions when a user’s immediate interest differs from their session context.
In real-world applications, the model has exhibited competitive performance that is particularly relevant for commercial settings. Its simplicity and interpretability, as well as efficiency, lend itself well for implementation in manufacturing settings. Show that attention-based models can effectively approximate dynamic user intent within sessions without recurrence or graph architecture.

2.2.2.6 Transformer-Based Methods

Another major development in session-based modelling has been the use of transformer architectures, which originated from Natural Language Processing research and are designed to capture long-range dependencies and bidirectional context.
BERT4Rec, proposed by Sun et al., suggests the use of the BERT architecture in sequential recommendation tasks. They replace a left to right sequence modelling with a bidirectional approach. According to the context around them, BERT4Rec predicts objects that are masked randomly in the sequence. This takes into account all past and future item dependencies, allowing for richer models of user behavior to be learned. In benchmark

datasets, BERT4Rec outperformed several state-of-the-art models. BERT4Rec's design also enables pretraining and scalability, which are two very important properties for transfer learning in recommender systems.

Moreira et al. (2021) expanded even further the potential of Transformer architectures in session-based recommendation by adding Transformers for Recommendation (Transformers4Rec), an open-source library and framework built on top of HuggingFace Transformers. They offer a very flexible range of session modeling possibilities, and they support several training paradigms that accommodate the great diversity of sessions. These training paradigms are Causal Language Modelling (CLM), Masked Language Modelling (MLM), Permutation Language Modelling (PLM), and Replacement Token Detection (RTD).

Transformers4Rec is known for permitting side data, such as user context and item characteristics, to be integrated, which enhances recommendation performance not just in news and e-commerce but also in other domains. It is also particularly good for next-click prediction activities where short-term interactions predominate. In these activities, Transformer-based models systematically beat both neural and non-neural baselines.

These Transformer-based methods complement the more general trend of using bidirectional attention and transfer learning to increase the scalability and resilience of session-based recommendation systems.

2.3 Auto Encoders for Recommendation Systems

2.3.1 Summary: Theory and Origins

Having first emerged in the early 2000s as tools for unsupervised representation learning and dimensionality reduction, autoencoders (AEs) represent a major development in machine learning, observed by Hinton and Salakhutdinov (2006). Two fundamental parts define the framework of AEs: a decoder reconstructs the original input from this more abstract representation and an encoder compresses input data into a latent space. For recommendation systems that handle sparse or high-dimensional user-item interactions, this approach helps AEs to find latent patterns and structures inherent in the data, so positioning them as especially successful (Vincent et al., 2008).

Their value also relates to the creation of concise user or session representations maintaining important preference data. AEs cleverly include behavioral patterns into latent embeddings by reducing reconstruction loss, so supporting tailored item recommendations (Sedhain et al., 2015). Furthermore, the denoising form of AEs improves resilience especially in partially observed and high-dimensional input spaces (Vincent et al., 2008).Autoencoders can be seen as a non-linear extension of conventional matrix factorization techniques; limited to linear activation functions, shallow AEs essentially replicate Principal Component Analysis (PCA), so highlighting their significance in producing interpretable latent component models.

2.3.2 Classical AE, Denoising AE, and Variational AE

Learning to replicate their input at the output, classical autoencoders produce a compressed latent representation. Though basic, they have shown great ability to capture latent preferences in group filtering projects (Sedhain et al., 2015).

Denoising autoencoders (DAEs) developed by Vincent et al. (2008) learn to replicate original input from a partially distorted version. This makes them more resistant to noise and more suited for implicit feedback data. The last one is a common feature in recommender systems where missing items often indicate user indifference rather than a lack of information. Usually implemented by randomly masking input bits, the corruption mechanism drives the model to generate stable representations insensitive to partial observation. DAEs theoretically map noisy input back to an underlying low-dimensional structure that captures important relationships, so enabling learning (Vincent et al., 2008).

Each input is encoded as a distribution across the latent space in a probabilistic framework offered by variational autoencoders (VAEs) (Jordan et al. 1999). Regularized with the Kullback-Leibler divergence, this probabilistic encoding lets VAEs show variation in user preferences and uncertainty. VAEs are especially helpful for simulating implicit interactions and handling cold-start situations since they learn to produce convincing data points unlike deterministic AEs (Liang et al., 2018). Combining Bayesian inference with reparameterization techniques preserves the advantages of a deep generative model while enabling effective training.

Recent research indicates that VAEs learn structured generative processes, hence they outperform DAEs in large datasets; but, DAEs may be more suited to smaller datasets where overfitting is a concern (Bennouna et al., 2022). Moreover, Ferreira et al. (2020) show that AE-based systems reduce data sparsity more successfully than conventional collaborative filtering techniques including Singular Value Decomision (SVD). SVD depends on linear approximations, thus when user-item matrices are quite sparse, this can be inefficient. Focus the benefits of neural-based collaborative filtering when handling quite sparse data matrices, their dropout-enhanced AE model has a low root mean squared error (RMSE).

AE-based recommender models have also expanded to include hybrid techniques. These techniques combine AEs with elements of collaborative or content-based filtering. Zhang et al. (2020) found two types of hybrid techniques. The closely linked approaches is when autoencoder representations directly influence recommendation scores. And the loosely coupled variations is when autoencoder pre-training is used to extract features for a different recommendation algorithm. Even in cold-start situations, efficiency has increased even more as a result of integrating side information into AE systems. This side information consists of user demographics and item descriptions. To improve feature robustness and training efficiency, new variants were added to the family of denoising autoencoders. They are:

- Stacked denoising autoencoders (SDAEs)
- Marginalization DAE (mDAE)

Liang et al. (2024) claim that VAEs stand out among deep learning-based recommenders due to four main traits:

- Their strong encoding capability for learning robust user and item representations;
- Their generative nature that supports data synthesis and preference simulation;
- Their probabilistic Bayesian formulation which allows uncertainty modeling and posterior regularization;

- Their flexible internal structure, which supports integration with other architectures like RNNs or CNNs and extension to various prior distributions.

These characteristics enable VAEs to more precisely replicate sparse, multimodal, cold-start environments than many other methods. VAE-based recommenders, for instance, have been developed to support multinomial, Bernoulli, or user-specific priors, so optimizing representations in line with observed behaviors.

2.3.3 Application to Session-Based Job Recommendations

SBRS which aim to forecast user behavior just based on actions performed during a single session, find particular fit for autoencoders. In the field of job recommendations, where consumers might not have past profiles or return for several visits, this is especially crucial (Lacic et al., 2020).

Lacic et al. (2020) proposed in session-based job recommendations a whole strategy using autoencoders to encode session data into latent representations. The most pertinent job ads are then found by cosine similarity analysis of these embeddings in a k-nearest neighbor system. They experimented three famous datasets in this field: Career Builder 2012, XING 2017 Challenge, and Studo Jobs.

Their results revealed that AE-based techniques offer not only competitive accuracy but also more originality and coverage, two important criteria in job recommendation where repeated ideas usually reduce user interest. Their method distinguishes itself in that it combines elements of job content and interaction data into the encoding process. Two variants are shown: one based on job description traits and one based on session-level click only. This hybrid design helps the system to operate effectively even in cases of regular change in job advertising. This issue causes instability in systems of pure cooperation. In terms of balancing relevance and variation, variational models performed especially nicely.

Their system design also takes scalability into account by means of dimensionality reduction and approximative closest neighbor search. The open-source publication of their architecture (Lacic et al., 2020) supports repeatability among the research community on recommender systems.

2.3.4 To Multimodal and LLM-Augmented Autoencoders

Although autoencoders effectively compress session data into pertinent latent vectors, they usually deal with ordered categorical inputs such as click sequences or item IDs. Rich unstructured data like job titles, descriptions, and skill tags does, however, often find place in job suggestion settings. Classical AEs cannot entirely use of all information.

Searchers are looking at using textual embeddings from big language models in AE-based recommenders to solve this. Beter generalization across job categories and sectors are achived from more in-depth semantic understanding of work content and user intent by these LLM-enhanced AEs (Sun et al., 2019).

Moreover, a good future will come by combining unstructured textual elements with structured interaction data in a multimodal AE framework.

Early or late fusion techniques may fit here. Such hybrid approaches open the path for more accurate and context-sensitive job recommendations by offering a more whole representation of both user behavior and item content.

2.4 Large Language Models in Recommendation

2.4.1 Introduction to Large Language Models

Large Language Models are deep learning architectures trained on large corpora of text with a human-like language understanding. Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) are widely applied for different NLP tasks. They are advantageous in recommendation systems with limited user data such as session-based settings since they can understand contextual and semantic messages. Yan (2023). The semantic modeling using LLMs supports personalized item matching even in cases where there is no historical user data.

According to Gedikli and Jannach (2026), systems that rely on item properties become more appropriate when those properties are semantically understood. By empowering systems with a better understanding of content, LLMs can make more relevant interpretations and recommendations by improving through the years and leveraging advances in NLP as well as access to structured world knowledge. The LLMs help in making recommendation systems that use both endogenous item data and exogenous data source to inform recommendations.

Semantics-based recommender systems depend on an awareness of item properties in a way that reflects their real meaning (Gedikli & Jannach, 2026). LLMs have greatly changed the way systems understand and suggest content as NLP and access to structured world knowledge advance. These systems support both endogenous (internal item data) and exogenous (external data sources like knowledge graphs) features to guide recommendations.

Recent observations imply that under LLM-augmentated models, recommendation will become ever more homogeneous with search systems. Yan (2025) said that LLMs nowadays support designs combining ranking pipelines with conventional vector-based retrieval (dense search). In cold-start scenarios, in which LLM-generated semantic IDs and content embeddings are used to prepopulate user and item representations, this is especially powerful.

Furthermore, big commercial uses like those of YouTube are starting to substitute LLM-derived embeddings created by residual vector quantization and transformer encoders (Yan, 2025) to replace static ID-based features. These compressed yet rich representations enable RS to change with new content and new users without entirely retrains.

LLMs enable fresh approaches of data generation. They serve to replicate realistic behaviors that support training ranking models in addition to fine-tuning models using synthetic query generation or user scenarios. LLM-assisted training was proven to speed cold-start adaptation and reduce bottlenecks in feature engineering pipelines (Yan, 2025).

2.4.2 Text Embedding with LLMs

Text embeddings are dense vector representations that capture semantic meaning, so reflecting natural language inputs such as user searches or job descriptions. Though often lacking deeper semantic nuance, traditional methods including TF-IDF and Word2Vec encode syntactic properties or co-occurrence statistics (Mikolov et al., 2013). Sentence-BERT (Reimers & Gurevych, 2019) among other LLMs produce contextualized embeddings that dynamically change to match word meaning depending on surrounding text. These model improved semantic capture has been helpful in job recommendations, where both job advertising and user intentions are captured as text (Cherifi, 2023). Studies like Liu et al. (2020), which apply semantic-based service descriptions for content filtering, demonstrating that LLM-based embeddings can outperform simply statistical similarity measures when context alignment is crucial, also help to support this. Closing the semantic-interaction gap typical in recommendation systems, BeeFormer (Vančura et al., 2024) shows even more how tuning LLMs with interaction data aligns embeddings more closely with user behaviors.

Particularly in job recommendations, clustering-based methods have been applied to preprocess textual job descriptions into structured feature representations LLMs can improve upon further. Mhamdi et al. (2020) for example grouped semantically related jobs using word embeddings and k-means, so clustering job offers. By matching user interaction data with contextualized job vectors, these clusters provide a semantic for downstream models, that is enabling personalized matching. Such clustering combined with LLM-based embeddings improves interpretability while maintaining semantic richness in recommendation outputs.

BeeFormer presents a training paradigm in which actual interaction data updates sentence Transformer models instead of depending just on static semantic similarity. This approach lets the model learn embeddings maintaining both semantic meaning and behavioral relevance. It solves the problem whereby two semantically similar objects could have rather different interaction patterns. Users frequently engage with accessory objects, for instance, that are behaviorally related but not textually like (Vančura et al., 2024).

Revealing latent patterns in session-based recommendation, Messica et al. (2017) showed how embeddings derived from techniques including Word2Vec and GloVe might represent user browsing sessions as "sentences" and items as "words." Their results in e-commerce environments revealed that recurrent neural networks employing item embeddings produced noticeably better predictions than conventional techniques such item-to--item similarity or matrix factorization.

SAID, Semantically Aligned item ID embeddings are proposed by Hu et al. (2024) in a two-stage architecture It shows even more how LLM-based embeddings might improve session-based recommendation. In Stage 1, a projector creates semantically aligned embeddings from item IDs by prompting an LLM the reproduce associated item descriptions. In Stage 2, the embeddings are then incorporated into lightweight sequential models like GRUs or Transformers. This means that there is no need for LLM inference at runtime, yet it maintains group-based semantic alignment learned training. Empirical evaluations suggest that SAID beats baselines on multiple public datasets, along with demonstrating production-grade latency.

Practical implementations allow several NLP-based methods to be stacked to enhance semantic matching and embeddings. For example, named entity recognition (NER) can extract structured entities (as companies and skills) from user searches or job advertisements, which LLMs can then use to generate more significant vectors. Furthermore helping with both precision and scalability (MobiDev, 2024) topic modeling and text summarizing methods let LLMs reduce informations into basic semantic representations.
Semantic-aware recommendation also gains from embeddings that capture domain-specific knowledge. LLMs enable generalization from textual semantics to item-level representations by means of embedding models such as Item2Vec or Doc2Vec (Le & Mikolov, 2014), so augmenting user-item match quality.
Still, difficulties persists. Lack of fine-grained labeled datasets often limits evaluation of embedding quality. And computational needs of LLMs cause scalability problems (Khan, 2023; Wang et al., 2024). Moreover, semantic embeddings produced from LLMs might not exactly reflect behavior-based preferences without further fine-tuning or adaptation techniques.

2.4.3 Measuring Semantic Similarity

Matching user preferences with content in recommender systems depends on semantic similarity estimation at least. Semantic relationships such synonymy or paraphrasing are not well captured by classical approaches including cosine similarity over TF-IDF vectors.LLM-derived embeddings model richer semantic relationships, so enhancing similarity detection across many language expressions (Yang et al., 2024).
As Liu et al. (2020) showed, semantic content-based methods use concept and logic-based models to examine service specifications. These techniques show that considering elements like inputs, outputs and preconditions enhances the precision of similarity matching.. By training item ID embeddings semantically aligned with textual descriptions, SAID (Hu et al., 2024) further advances this and enhances downstream performance in session-aware models. Semantic alignment and contextual modeling shows the degree of personalization that LLM-based similarity measures allow.
LLM-powered embeddings directly solve the limits of previous vector space models, which battled synonymy and polysemy. LLMs are a fundamental tool for contemporary recommendation systems since, as Gedikli and Jannach (2026) point out, they provide context-sensitivity and linguistic complexity to semantic similarity scoring.

2.4.4 Enhancing Personalization with LLMs

Often in session-based recommendations, personalizing them relies on simulating transient user behavior. LLMs improve this by precisely capturing user intent in complex ways even in cases of little interaction history. Transformers such as BERT4Rec (Sun et al., 2019) for example model bidirectional dependencies in user sessions and offers strong next-item predictions.
Providing a useful method for embedding personalization, SAID as viewed above (Hu et al., 2024) convert item IDs into semantically meaningful representations that fit easily with lightweight sequential models such as GRU or Transformer. By means of fine-tuning these representations to reflect the underlying text, we enable enhanced inference performance and personalization free from the computational overhead of involving the LLM at inference times.
Multi-modal and multi-source data fusion benefits LLM-based personalizing as well. Rich modeling of user preferences is achieved, for instance, by including structured metadata, unstructured job descriptions, and

behavioral signals into unified LLM-based architectures (Yan, 2023; Liu et al., 2020). Training LLMs with both textual side information and user interaction data helps to produce accurate recommendations even in cold-start or zero-shot conditions, according to the beeFormer framework (Vančura et al., 2024).
Furthermore quite common are hybrid recommender systems combining content semantics derived from LLMs with collaborative filtering. Gedikli and Jannach (2026) claim that these systems learn from side data to improve performance in sparse datasets and provide a good reaction to cold-start constraints.

2.4.5 Limitations and Future Challenges

Even with major progress, including LLMs into recommender systems has certain challenges:

- Scalability: The computational complexity of LLMs, especially in real-time applications, presents challenges. Efficient architectures and compression techniques, like LoRA or retrieval-augmented generation (RAG), are needed to mitigate inference costs (Wang et al., 2024).
- Cold-Start Alignment: Although LLMs improve cold-start recommendations, aligning semantic content with behavioral relevance requires further exploration. Hybrid training frameworks like SAID show promise but require careful optimization to avoid performance regressions.
- Interpretability: LLMs often act as black boxes, which complicates explanation and transparency in recommendation (Zhang et al., 2024). Embedding alignment methods that preserve traceable semantic features may offer a partial remedy.
- Evaluation: Standard accuracy metrics may not fully capture the improvements brought by semantic embeddings. Beyond-accuracy measures like novelty and diversity are essential to assess the holistic impact of LLM-enhanced models (Lacic et al., 2020).

With their capacity to model context, semantics, and user intent, LLMs greatly better the performance of session-based recommenders. Their use in generating text embeddings, computing semantic similarity, and enabling dynamic personalization allows more precise recommendations. Recent developments such SAID show the possibilities of merging LLMs with interaction data to produce behaviorally aligned representations that enhances performance even in sparse-data or cold-start conditions. Scalability, explainability, and evaluation techniques should be among the topics of future studies to guarantee responsible and efficient application of these models in practical recommendation systems.

2.5 Multimodal Fusion Techniques

Multimodal data fusion has become a fundamental technique in improving job SBRS. To do so, it has allowed it to combine heterogeneous data types including textual content, user interaction logs, and visual information. Richer user and item representations made possible by this integration can greatly increase the accuracy, personalization, interpretability of recommendations (Gaw et al., 2022; Zhang et al., 2023).
With the increasing complexity of deep learning models, this classification becomes less suitable even if conventional taxonomies classify fusion into early, representation-level, and late strategies. Modern designs progressively combine decision-making, fusion, and representation learning into a single pipeline (Zhao et al., 2024.).

2.5.1 Levels of Multimodal Fusion

Depending on the stage of integration between data from several modalities, multimodal fusion techniques are arranged into three levels.

2.5.1.1 Feature-Level Fusion (Early Fusion)

Before they are processed by the model, raw or low-level features from several modalities are combined under feature-level fusion. For example combine textual embeddings from job descriptions with visual embeddings from logos or banners into a single input vector. This lets the model learn direct cross-modal interactions during training (Baltrušaitis et al., 2019). Early fusion techniques, however, may suffer from scalability problems when working with high-dimensional modalities (Liu et al., 2024) and often call for synchronized and aligned data.

2.5.1.2 Representation-Level Fusion (Hybrid Fusion)

Fusion techniques at hybrid or representation-level combine early and late fusion techniques. These methods learn separate modality-specific representations, then aggregated using gating, neural networks, or attention mechanisms. A session-based job recommender might, for instance, use CNNs for image data and RNNs for clickstream data later merging the outputs via an attention layer to create a joint representation (Yu et al., 2020). It is particularly helpful in session-based systems where interactions are sequential and context-sensitive.

2.5.1.3 Decision-Level Fusion (Late Fusion)

Decision-level fusion compiles the output predictions from models taught separately on every modality. The final recommendation is synthesized using methods including weighted averaging, meta-learning, or voting. Though it may underexploit deeper cross-modal interactions (Zhang et al., 2022), this modularity makes late fusion easy to implement and robust to missing modalities.

2.5.1.4 Historical and Cross-Modality Insights

The fundamental framework developed by Ailyn (2024) helps one to further contextualize multimodal fusion techniques. It offers a domain-agnostics summary of early, late, and hybrid fusion techniques. Taxonomy emphasizes the trade-offs involved as well as the significance of every level's function. Ailyn (2024) points out, for example, that early fusion presents problems with data alignment and noise propagation even it allows the joint learning of features across modalities. Late fusion reduces modality-specific modeling but may lose cross-modal interactions. Though difficult to design, hybrid fusion is acknowledged for its adaptability to the data structure and learning goal.
Morover, Ailyn (2024) highlights the fact that ensemble learning plays in fusion pipelines, which might be included into job recommender systems to increase resilience. Very important factors for real-time SBRS deployment are also computational complexity, missing modalities, and model interpretability.
This point of view clarifies the knowledge that a fusion technique can’t fits all situations. Systems should be customized by assessing data quality, application latency, and model explainability needs. Deep learning techniques, ensemble learning, and robust feature representations taken together define the changing terrain of multimodal data fusion.

Especially one of the first and most important studies in multimodal deep learning by Ngiam et al. (2011) presents important ideas still applicable today: shared representation learning and cross-modality learning. These concepts have changed the way models handle imbalances and data shortage across several modalities. Their deep autoencoder architecture shows that models trained with access to paired modalities (like audio and video) can use shared embeddings to improve representation learning even when only one modality is available at inference. For job recommender systems, where behavioral signals or metadata might be lacking in some user sessions, this is especially pertinent. Moreover, their findings reveal that training on noisy or missing modality inputs can improve robustness, which is important for session-based systems functioning in real-world, multimodal environments.

2.5.2 Techniques and Architectures for Fusion

2.5.2.1 Autoencoder-Based Fusion for Session Representations

Built totally around autoencoder architectures, Lacic et al. (2020) propose a session-based job recommender framework. These models learn to encode user sessions into latent embeddings then use k-nearest neighbor search across the latent space to generate job recommendations. They assess two approaches of input modeling. One depending just on session interactions and another combining interactions with binary-encoded job content features. Specifically in terms of beyond-accuracy measures like novelty and coverage, their results show that variational autoencoders (VAE) routinely achieve competitive or superior performance across three real-world datasets.

In the framework of the job recommendation, the suggested architecture is especially beneficial. This is so because it can run under session-level anonymity. Including content-based signals will also help to address cold-start and item sparsity problems. Choices about input modeling (interaction-only vs. interaction + content) impact latent space clusterering patterns, and hence recommendation performance. Furthermore practical for production deployment is the use of a fixed-length vector representation via content padding, which lowers the demand for regular model retraining.

2.5.2.2 Transformers and Post-Fusion Context Layers

Moreira et al. (2021) offer a successful implementation of multimodal Transformer architectures especially tailored for session-based recommendation in e-commerce. Transformer-XL and XLNet, trained respectively with causal and masked language modeling goals, are combined in their ensemble method. Especially, they suggest a post-fusion context layer called Latent Cross, which improves prediction by combining session-level search signals with tabular, textual, and image features. This post-fusion mechanism essentially conditions predictions on dynamic user context by multiplying the core prediction vector with a context vector generated from session-specific data.

Their approach also introduces custom features such time-of-day and recency signals and uses frequency capping to handle long-tail objects, so illustrating how multimodal inputs can be enriched for higher resolution modeling. They preserve inter-item semantic similarity by applying L2 normalisation to pre-trained text and image embeddings, so supporting next-click prediction. In a major session-based recommendation competition (SIGIR eCom 2021), these architectural elements produced top-ranking performance.

Job recommender systems, where user behavior is session-limited, items are multimodal (job descriptions, logos, …), and cold-start issues are common, especially benefit from this implementation. The Latent Cross method can be adjusted to include user session search terms and recently viewed job posts by the user to make user-context aware predictions.

2.5.2.3 Taxonomies of Fusion Strategies

Liu et al. (2024) divide fusion strategies into four categories: Modality Encoder, Feature Interaction, Feature Enhancement, and Model Optimization. Major issues in multimodal recommendation including representation alignment, data sparsity, modality imbalance, and computational efficiency are addressed in their framework. The authors underline how domain-specific encoders like ViT for images and BERT for text, may be combined with item IDs to produce consistent feature representations in modality encoding. Scaling SBRS models using both structured and unstructured job data depends on this division of modality-specific and identifier-level encoding.

Regarding feature interaction, Liu et al. find three primary mechanisms: Bridge, Fusion, and Filtration. Bridge-based methods such as those based on graph neural networks catch item-item and user-item interactions across several modalities. While filtration systems remove noisy modality-specific data that may compromise recommendation accuracy, fusion techniques combine different modality signals. These techniques complement approaches already used in job RS to harmonize textual and behavioral signals.

Finally, Liu et al. underline the need of model optimization by separating two-step and end-to- end training pipelines. Industrial deployment depends on this difference since real-time performance and computational cost count. Practical ways to keep high performance without using much resources are fine-tuning pretrained encoders or distilling simpler models.

2.5.3 Applications in Session-Based Job Recommendation

In the context of job recommendations, multimodal fusion lets unstructured data (such as job descriptions, resumes, company reviews) be combined with structured elements (such as job location, industry, salary). Models can better predict user intent by including behavioral signals including click patterns, dwell time, and scroll behavior, so refining user profiles (Zhao et al., 2023).
Job descriptions and user inquiries, for example, can be represented using embedding methods such as BERT, which then hybridly fuses with interaction histories. This not only enhances cold-start performance but also facilitates the dynamic shifting of user intent across sessions, so addressing the difficulty.
Moreover, especially when combined with other modalities during early or representation-level fusion, visual information such as company logos or office images can improve the user experience and engagement.
Furthermore providing flexible implementation of multimodal recommendation in job platforms is the autoencoder-based method of Conceição et al. (2019). Especially in sparse environments typical of recruitment domains, their method shows that unsupervised learning from video data (like trailers, company intros) and job metadata can greatly improve cold-start and top-N prediction tasks.
Using multimodal fusion within session-based job recommendation, Lacic et al. (2020) provide a specific and concrete application with regard to domain. They show how well autoencoders encode content-based features from job descriptions as well as behavioral interaction data. They demonstrate that fixed-length session representations made possible by combining content and interaction data help to lower retraining requirements while yet preserving contextual relevance. This is especially helpful for production systems with high user anonymity and fast item turnover, such as job platforms..
Their use of three real-world datasets validate the generalizability of multimodal fusion strategies across user and job pools with different sparsity by demonstrating that content-augmented latent session vectors offer better personalization under cold-start conditions. a specific, concrete application of multimodal fusion within session-based job recommendation.

2.5.4 Benefits of Multimodal Fusion

For session-based job recommender systems (SBRS), multimodal data fusion presents a broad spectrum of advantages covering performance, user experience, architectural robustness.

From a performance and personalizing perspective, multimodal fusion greatly improves models' capacity to build rich and complex user profiles. By including different user behaviors and item content, techniques such as modality-aware embeddings help to enable enhanced personalization Fusion models also show strong noise resistance since overlapping and complementary modalities can fill in for missing or contradictory signals. Moreover, by projecting high-dimensional and sparse item data into dense latent spaces, architectures including autoencoders efficiently lower feature sparsity. This enhances model generalization and makes more stable training easier as well.

Regarding user-facing effect, multimodal fusion enhances the capacity of the system to provide more context-aware and informed recommendations. Improved interpretability lets practitioners and users know how various data types affect the output at last. Furthermore, these systems get more originality and diversity in recommendations by including content modalities including text, images, and behavioral traces. This helps reduce the risk of popularity bias and raises the possibility of revealing underexplored employment prospects, so improving user satisfaction and involvement.

At last, on the architectural and operational side, multimodal fusion supports scalability and generative capacity. Those qualities are great for practical application. Modern architectures fit fast-moving job platforms since they can scale with changing data sources and session patterns, allowing transformers and graph neural networks to be fit.

Beyond these overall advantages, Lacic et al. (2020) offer empirical data showing higher recommendation novelty and session coverage resulting from multimodal fusion using autoencoder-based embeddings. On job platforms, user satisfaction depends critically on this. Their system finds hidden job options by encoding latent features from both job content and interaction patterns, so reducing popularity bias and enabling users to investigate a wider item space. Derived from categorical job attributes (location, discipline, …), the fixed-size binary feature vectors also simplify scalability in practical implementations. Their work confirms that, particularly when variational autoencoders are combined with session similarity-based ranking techniques, novelty and diversity can be obtained without sacrificing accuracy. By encouraging diversity and so reducing echo chambers, multimodal fusion not only increases recommendation accuracy but also user engagement.

2.5.5 Challenges and Future Directions

Session-based job recommender systems still have several structural and pragmatic difficulties even if multimodal fusion offers many benefits. Data sparsity and imbalance are a major problem especially in situations when users interact just briefly, which makes it challenging to find significant trends. Furthermore, the combination of heterogeneous modalities (text, images, and behavioral signals) presents a challenging work because of variations in temporal scale, granularity, and dependability. It is still rather easy to align these modalities into coherent representations.
Using their autoencoder-based job recommendation system, Lacic et al. (2020) draw attention to a number of these constraints. Especially, they highlight the overhead brought about by updating job IDs in sparse binary input vectors, which can cause pipeline inefficiencies and retraining loads. Their method revealed latent space clustering risks connected more with session length than with user intent even using content-based fixed-length vectors. These realizations highlight the need of more adaptable, intent-aware models of input. Furthermore underlined by them are the fact that benchmark datasets sometimes overlook beyond-accuracy indicators, which are absolutely vital for assessing models in the framework of actual hiring systems.

Chapter 3 - Methodology

Content:

3.1 Research Philosophy and Design

This research adopts a positivist philosophy, emphasizing the quantitative evaluation of model performance. It follows a deductive approach, testing predefined hypotheses concerning the effectiveness of session-based job recommender systems enhanced by autoencoders and BERT embeddings.

The methodological choices are grounded in the findings and taxonomies discussed in the literature review. The identified challenges in job recommender systems, particularly session-level anonymity, cold-start problems, and semantic misalignment, directly inform the system design. Specifically, this study builds upon the foundational work of Lacic et al. (2020), who proposed session-based autoencoder architectures using binary-encoded item interactions. Their findings on recommendation quality, novelty, and coverage set a relevant benchmark.

To address limitations highlighted in their work, such as the absence of semantic content modeling, this research introduces enhancements using BERT-based job embeddings. The methodological design thus integrates deep learning, session-awareness, and multimodal fusion as motivated by the literature, aiming to improve the richness of session representation and recommendation diversity.

The study employs a descriptive and explanatory design, systematically comparing different autoencoder architectures. A cross-sectional time horizon is applied, evaluating models using static datasets.

3.2 Data Collection and Preprocessing

3.2.1 Dataset Description

The pipeline is based on Lacic et al. (2020).

The CareerBuilder 2012 dataset was utilized, a publicly available, large-scale dataset from Kaggle used as a benchmark for job recommendation tasks. From this dataset, only two files were used:

- jobs.tsv: Contains information about job postings. Each row describes a job post with a unique JobID, its associated WindowID, and additional descriptive features. State, City, Country, Zip code, Title, Description and Requirements are textual fields. Notably, StartDate and EndDate columns specify the visibility window of the job on the platform. Users could only apply to a job within this timeframe, making it essential for correct interaction modeling.
- apps.tsv: Records applications submitted by users to jobs. Each entry specifies one application with the UserID, JobID, application WindowID, Split, and the ApplicationDate.

However, Lacic in the need to match that dataset to others, only used few columns and decided to create session based on an other strategy. Data preprocessing involved several steps.

3.2.2 Application Log Processing

Regarding the application log file:

- Sessions were defined by assigning session IDs based on a 30-minute inactivity threshold. Events were sorted by user ID and timestamp, and session IDs were incrementally assigned whenever the time exceeded 30 minutes or a new user appeared. This sessionization process ensured that each session represents a continuous period of user activity.
- The training and test sets were constructed using a time-aware strategy based on session end times. The last_n_days_out_split method was applied:
  - Sessions whose last interaction occurred in the final day of the dataset timeline were assigned to the test set.
  - All earlier sessions were used for training.
  - To ensure test reliability, items not present in the training set were filtered out from the test set.
- Sessions with fewer than three interactions were removed to maintain contextual relevance.

3.2.3 Job Description Processing

Regarding the job description file:

- Textual content from job titles, descriptions, and requirements were lower cased and cleaned through multiple regex operations to remove HTML tags and unwanted characters.
- To embed the cleaned textual content, the all-MiniLM-L6-v2 model from the Sentence-Transformers library was used. This model produces dense 384-dimensional vector representations optimized for semantic similarity and sentence-level understanding. It is particularly effective for short text like job titles and descriptions, and produces embeddings that are compact, semantically rich, and computationally efficient, making them ideal inputs for autoencoders. This replaced the one-hot encodings of the retrieved topics using LDA that was set up in the original paper.
- Like Lacic's pipeline, only the State column was used as one-hot-encoded.

3.2.4 Session Encoding and Input modeling

To further enhance the representation of job postings, we construct a hybrid input vector that combines both structured and semantic content features. Session encoding was generated from job description file and application log file. Each job representation was concatenated to the other job representation from the same session. This concatenation resulted in a 2D array of shape [number_of_sessions, item_vector_size × max_num_items].

A job vector, or item vector, is made of the embbeiding columns from BERT and the one-hot-encoded states. Bert output a 364 dimensions embeddings for each of the 3 textual features. There are 55 different states. Hence, one item vector is 1199 features.

The session vectors are limited to 25 jobs. Hence one session vector is 29975 features. If a session don't actually reach 25 jobs, the session vector is right padded with 0.

Previous work was based on one-hot-encoded vectors. Since this experiment uses BERT embeddings, this results in arrays of continuous floating-point values. This is why feature standardization using StandardScaler was necessary. This transformation standardizes the features by removing the mean and scaling to unit variance. It helps stabilize training with optimizing the gradient descent process and preventing numerical instability. It also ensures that the autoencoder treats all features with equal importance.

3.3 Model Architecture and Session Encoding

Each autoencoder implemented in this study follows the theoretical formulations outlined by Lacic et al. (2020) and was adapted to work with continuous BERT-based input embeddings.

Those autoencoders were trained with to reconstruct the embedded item vectors. Only the encoder is used is the next step of the experiment. The decoder is here to ensure the latent space is coherent.

3.3.1 Classic Autoencoder (AE)

As introduced by Kramer (1991), autoencoders reduce the dimensionality of data while preserving structure more effectively than PCA. The AE learns a compressed latent representation by minimizing the mean squared reconstruction error through a symmetric encoder-decoder architecture. The encoder maps an input vector to a latent representation via a non-linear transformation. During inference, this latent vector is used to represent the session. ReLU was used as the activation function for all hidden layers and Tanh for the output.

3.3.2 Denoising Autoencoder (DAE)

As shown by Vincent et al. (2008), DAEs extend classical autoencoders by corrupting the input to encourage robustness in the learned representation. In this study, additive Gaussian noise was applied with a corruption probability of 0.5. This helps the network learn representations resilient to noise or missing data. The architecture and activation functions remain consistent with the AE.

3.3.3 Variational Autoencoder (AE)

The VAE models the latent space using variational inference and samples latent representations via the reparameterization trick. The encoder consists of a dense layer followed by batch normalization and dropout. It produces both a mean vector and a log-variance vector. To ensure numerical stability and prevent extreme values in the estimated variances, the log-variance vector is clipped to a bounded range. This step mitigates potential issues such as gradient explosions, overconfident posterior approximations, or NaNs in the KL divergence calculation. A custom sampling function draws from a standard Gaussian using the reparameterization: $z = \mu + \sigma \cdot \varepsilon$, where $\varepsilon \sim N(0, I)$. The decoder mirrors the encoder structure using a sequential model with ReLU activation, batch normalization, and dropout, followed by a dense output layer with Tanh activation.

A custom loss layer combines the reconstruction loss (mean squared error) and KL divergence, scaled by a weighting factor. This design provides more control over the influence of regularization during training. The model was trained with the RMSProp optimizer and early stopping, and both encoder and decoder were saved separately for downstream use.

3.4 Analytical Procedures: Generating Recommendations

3.4.1 Modeling Session Vectors

Only Variant 2 (Content-based) modeling strategy from Lacic's experiment is implemented here. As exposed above, sessions are encoded by concatenation of different embedding from the textual fields of the job postings. This approach ensures fixed-length input vectors, thus reducing the necessity for frequent model retraining due to new job postings.

The autoencoders are trained to reconstruct the embedded item vectors using unsupervised learning. While the full autoencoder is optimized during training, only the encoder is retained for the next stage of the experiment. The decoder's role is primarily to enforce a coherent and structured latent space through the reconstruction objective.

3.4.2 Similarity-Based Ranking Mechanism

To generate recommendations, we first compute the latent session embeddings zs using the trained autoencoder encoders. For each target session st, we infer its latent representation and identify its most similar past sessions using cosine similarity. To reduce computational complexity, we restrict candidate sessions to those where users interacted with at least least one job in st.

Once the top-k most similar sessions are retrieved, job scores are computed based on how frequently each job appears in those sessions. Top-n jobs are ranked according to the following scoring function:

$$sKNN(s_t, j_i) = \sum_{i=1}^{n} sim(s_t, s_i) \times 1_{s_i}(j_i)$$

where sim(st,si) is the cosine similarity between the target session and a candidate session si, and 1si(ji) is an indicator function equal to 1 if job ji appeared in session si, and 0 otherwise.

In other words, the knn approach get k most similar sessions. For each jobs within those k sessions, the algorithm sum the similarity score of each session in which this specific job appears. Then, the function return the top n jobs, sorted by score.

3.5 Evaluation Framework

To assess the performance of the proposed session-based job recommender models, both accuracy and beyond-accuracy metrics were employed. These metrics were selected to capture not only how well the system predicts the correct items but also how novel, diverse, and inclusive the recommendations are.

3.5.1 Accuracy Metrics

nDCG@k (Normalized Discounted Cumulative Gain): This metric measures the quality of the ranking of recommended items. It penalizes relevant items that appear lower in the list, emphasizing the importance of top-ranked results. It is especially useful in recommendation tasks where the position of relevant items matters.

MRR@k (Mean Reciprocal Rank): This measures the inverse rank of the first relevant item in the recommendation list. It is sensitive to whether the correct item appears early in the ranking, rewarding recommendations that place relevant items at the top.

3.5.2 Beyond-Accuracy Metrics

EPC (Expected Popularity Complement): This metric evaluates novelty by assigning higher scores to recommendations that include less popular items. A higher EPC indicates that the model avoids over-recommending popular jobs, which can help users discover more diverse opportunities.

EPD (Expected Profile Diversity): This metric measures how different the recommended items are compared to the user's past interactions in the same session. Higher EPD values indicate more diverse and less redundant recommendations.

Item Coverage@k: The proportion of unique items recommended across all sessions. This metric provides insight into how broadly the recommender explores the item space.

Session Coverage@k: The percentage of sessions for which the model was able to generate at least one recommendation. This metric reflects the general applicability and robustness of the model.

The formulas are in appendix.

It's worth noting that @k refers here to the number of job recommendations.

These metrics were computed on the test set produced via the time-based session split.

In this work, we adopt the same evaluation framework, enabling a direct comparison of results. The top-k recommendations generated by the kNN procedure were compared to the actual next interactions observed in each session.

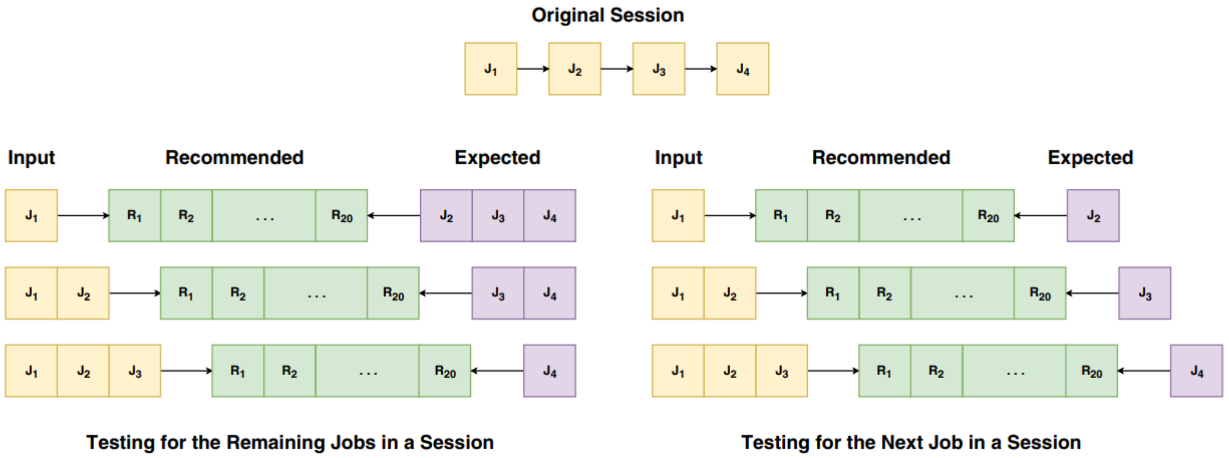

Original Session
J1
J2
J3
J4
Input
Recommended
Expected
R1
R2
...
R20
Testing for the Remaining Jobs in a Session
Testing for the Next Job in a Session

Figure 1 - Two different methods to test the session based job recommender system (Lacic)

Because the only modification in this study concerns the use of BERT-based session representations, we did not retrain the interaction-based variant (Variant 1) or replicate the baseline models originally tested by Lacic et al. (2020). Instead, we leverage their results as benchmarks, since the experimental setup, evaluation metrics, and dataset are aligned.
This approach allows us to isolate the impact of richer content embeddings on recommendation performance, while maintaining comparability with prior work. The top-k recommendations generated by the kNN procedure were compared to the actual next interactions observed in each session. Performance was evaluated separately for each autoencoder variant (AE, DAE, VAE) to identify strengths and trade-offs in accuracy, novelty, and diversity.

3.6 Main Differences from Lacic et al. (2020)

This study preserves the core experimental design introduced by Lacic et al. (2020), including the dataset (CareerBuilder 2012), evaluation metrics, and recommendation logic, but introduces several targeted modifications aimed at enhancing content modeling and compatibility with continuous vector representations.

The primary difference lies in how content is encoded. While Lacic et al. used one-hot encodings derived from topic modeling, this work substitutes those with BERT-based embeddings of job titles, descriptions, and requirements. These embeddings capture deeper semantic structure and richer linguistic features.
To handle continuous-valued BERT vectors, architectural modifications were made to the original autoencoders. The output activation was switched from sigmoid to linear (AE, DAE) or Tanh (VAE), and standardization using StandardScaler was applied to ensure numerical stability. Additionally, the loss function for AE and DAE was changed from Kullback–Leibler divergence (as used by Lacic et al.) to mean squared error (MSE), which is more appropriate for reconstructing continuous, real-valued vectors rather than binary inputs. For the VAE, we retained the variational framework and used a custom loss layer that combines the MSE reconstruction loss with the KL divergence term, scaled by a tunable weight. To further stabilize training, we clipped the log-variance output of the encoder to a fixed range before computing the KL term. This prevents numerical instability, especially when variances become extremely small or large. This adaptation maintains the generative nature of the VAE while ensuring compatibility with continuous embeddings and robust latent space learning.

Standardization Justification: Standardizing the input was crucial due to the floating-point nature of BERT embeddings. Without it, large unnormalized values would lead to unstable gradients and biased learning. In VAEs, it helped prevent erratic latent space distributions and made KL divergence regularization more meaningful. It also ensured that the Tanh output layer reconstructed values within a realistic range.

Due to hardware limitations, this study focuses on a subset of the whole dataset.

Evaluation Protocols: Following Lacic et al., two test setups were adopted: (1) predicting all remaining jobs in a session, and (2) predicting only the next job. This design ensures consistency with prior work and supports reproducible comparison.

While some architectural experiments like attention mechanisms were tested in Lacic's work and showed no major gains, this study did not re-explore them, maintaining focus on the impact of embedding choice and input normalization.

3.7 Validity, Reliability, and Ethical Considerations

Reliability was ensured through consistent preprocessing and model evaluation protocols across multiple iterations. Validity was strengthened by benchmarking against established baseline models (e.g., GRU4Rec, session-KNN) and employing multiple evaluation metrics (nDCG, MRR, EPC, EPD, coverage). Ethical compliance was assured through the use of publicly available and anonymized datasets, aligning with ethical standards in research.

Chapter 4 - Experiments and Discussion

Content:

4.1 Research Hypotheses and Expectations

Before presenting the experimental results, we outline the expectations that guided this study’s design and model evaluation. Based on theoretical insights and empirical evidence from Lacic et al. (2020), the following hypotheses were formulated:

H1 (Semantic Encoding Improvement):
Autoencoders trained on BERT-based embeddings of job content are expected to outperform their one-hot topic-model counterparts in terms of semantic novelty and recommendation diversity. These improvements are to be reflected by higher EPC (Expected Popularity Complement) and EPD (Expected Profile Diversity) scores.

H2 (Accuracy Preservation):
Despite shifting from sparse binary inputs to dense continuous BERT embeddings, the new models (AE, DAE, VAE) are expected to maintain comparable accuracy to those reported by Lacic et al., as measured by nDCG@k and MRR@k.

H3 (Expressiveness of Variational Autoencoders):
As shown by Lacic et al. (2020), VAEs demonstrated superior performance in beyond-accuracy metrics, notably in system-based and session-based novelty. Thus, we expect the VAE variant to produce the most expressive and semantically rich recommendations, reflected by the highest EPC and EPD scores among the three models.

H4 (Robustness of Denoising Autoencoders):
Lacic et al. also highlighted the stability of DAEs under noisy or sparse inputs. Therefore, the DAE is expected to exhibit robustness, measured by relatively consistent nDCG and MRR performance across varying session lengths and sparsity levels, even if it does not achieve the highest overall novelty.

These hypotheses guide the analysis of whether integrating BERT-based semantic embeddings into autoencoder frameworks enhances recommendation quality in session-based job recommendation tasks.

4.2 Experimental Setup

This section details the experimental configuration used to evaluate the impact of BERT-based semantic embeddings on session-based job recommendation models. The goal is to isolate the effect of richer content embeddings within the autoencoder-based recommendation framework introduced by Lacic et al. (2020).

4.2.1 Dataset Overview

The experiments were conducted on the CareerBuilder 2012 dataset, a large-scale benchmark dataset comprising job applications. After preprocessing:

- Job postings: 21532
- User sessions: 5016
- Interactions: 27087
- Users : 3992

4.2.2 Session Construction

Sessions were created using a 30-minute inactivity threshold. Only sessions with three or more job applications were retained. A time-based split assigned the final day of interactions to the test set, while the remaining sessions formed the training set. Job postings not appearing in the training set were filtered from the test set to ensure realistic cold-start conditions.

4.2.3 Feature Engineering

Each job was represented by a hybrid vector combining:

- BERT embeddings (384 dimensions each) for the job title, description, and requirements, for a total of 1,152 dimensions.
- One-hot encoding of the job's state (50 dimensions).

4.2.4 Final Input Vector Dimensions

This yielded a 1202 -dimensional item vector. Sessions were represented by concatenating up to 25 such item vectors, resulting in a 30050-dimensional session vector. Shorter sessions were right-padded with zeros.

4.3 Model Configuration and Training

4.3.1 Shared Model Architecture

Three autoencoder variants were evaluated:

- Autoencoder (AE): Standard feedforward encoder-decoder minimizing reconstruction loss (MSE).
- Denoising Autoencoder (DAE): AE with additive Gaussian noise (50% probability) to improve robustness.
- Variational Autoencoder (VAE): Incorporates probabilistic encoding using mean and log-variance layers; latent vector sampled via reparameterization trick. Loss = MSE + scaled KL divergence.

All models used the following symmetric architecture:

- Input layer: 30050 units
- Encoder: Dense(256) → Dense(100) with ReLU activations
- Latent space: 100 units (AE/DAE), mean and log-variance layers (VAE)
- Decoder: Dense(256) → Dense(30050)
- Output activation: Tanh (for continuous input reconstruction)

The VAE's KL divergence term was scaled with a tunable factor to balance expressiveness and reconstruction quality. Log-variance was clipped to a bounded range to ensure numerical stability.

4.3.2 Training

All models were trained using the RMSProp optimizer. The loss function was mean squared error (MSE) for AE and DAE, and a combination of MSE and KL divergence for the VAE. This differs from the original implementation by Lacic et al. (2020), which used a Kullback–Leibler divergence-based loss suited for binary one-hot encoded input. Because this study used dense, continuous input vectors derived from BERT embeddings rather than sparse binary encodings, the MSE loss was more appropriate to capture the magnitude of reconstruction error in a continuous feature space.

To further stabilize training, L2 activity regularization (1e-4) was applied to all dense layers. This constraint discourages large weights, which could arise from high-dimensional input features, and helps prevent overfitting. For the VAE model, the log-variance outputs were clipped to a fixed range to prevent numerical instability such as exploding gradients or undefined KL terms, which are common when working with continuous-valued embeddings. No dropout was used in AE and DAE architectures. To improve robustness in the DAE, Gaussian noise with a standard deviation of 0.5 was added to the input vectors during training. Early stopping was implemented by monitoring the validation loss with a patience threshold of five epochs.

Only the encoder was retained after training for downstream recommendation. The decoder was used exclusively for learning a coherent latent space.

During training, the AE was trained for five epochs, with validation loss initially decreasing from 16.76 to 9.67 before fluctuating. This pattern indicates that while the model was able to capture useful structure, it may have been limited in capacity or overly sensitive to noise, resulting in moderate overfitting. The DAE, trained for only three epochs, exhibited fast training loss reduction, but its validation loss increased from 12.63 to 14.04, suggesting overfitting and a lack of robustness despite the introduction of input noise. This may be due to the continuous nature of BERT embeddings, where additive Gaussian noise disrupts important semantic structure. In contrast, the VAE trained for 20 epochs showed a clear, steady decline in both training and validation loss, dropping from 237 to 214 and from 316.5 to 230.3 respectively. The VAE's probabilistic regularization, combined with clipping of the log-variance and the suitability of MSE for continuous input, enabled it to learn stable, generalizable representations.

4.3.3 Hardware Constraints

Due to hardware limitations, a subset of the full CareerBuilder dataset was used. Nonetheless, data volume remained sufficient to evaluate performance trends and validate the proposed enhancements.

4.4 Exploratory Data Analysis (EDA)

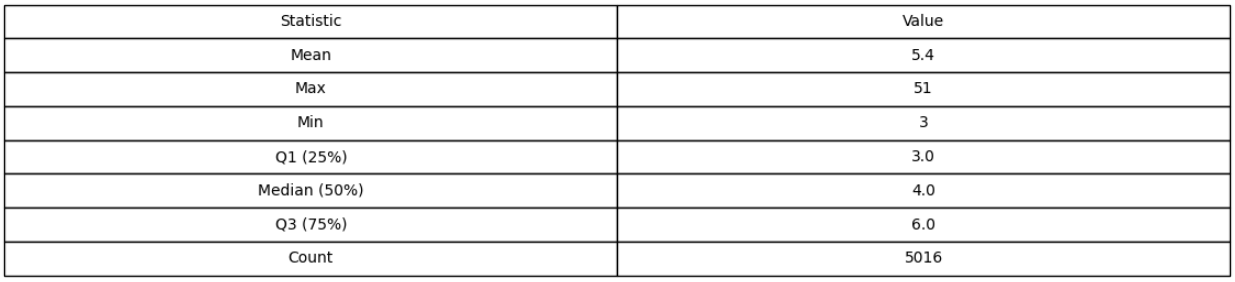
This exploratory analysis revealed several important characteristics of the dataset that influenced downstream modeling decisions. The short length of most user sessions, coupled with the highly sparse item interaction distribution, highlights the necessity of session-based modeling and the relevance of novelty-aware evaluation. The state distribution confirmed the need for one-hot encoding to handle regional bias, while the BERT embeddings proved to be semantically rich and highly unique. Overall, the dataset's sparsity, imbalance, and content variability reinforce the importance of using robust, diversity-oriented, and semantically informed recommendation techniques.

4.4.1 Session Characteristics

Sessions were constructed using a 30-minute inactivity threshold, resulting in a total of 5,016 user sessions included in the training and test sets. The interaction count per session ranged from 3 to 51, with a mean of 5.4. The distribution was right-skewed:

- Q1 (25%): 3 interactions
- Median (50%): 4 interactions
- Q3 (75%): 6 interactions

Table 1 - Basic statistics for the interactions per session feature



Statistic	Value
Mean	5.4
Max	51
Min	3
Q1 (25%)	3.0
Median (50%)	4.0
Q3 (75%)	6.0
Count	5016

Over 75% of sessions included six or fewer interactions. Sessions with fewer than 3 interactions were discardedduring the preprocessing to ensure meaningful context for session encoding.

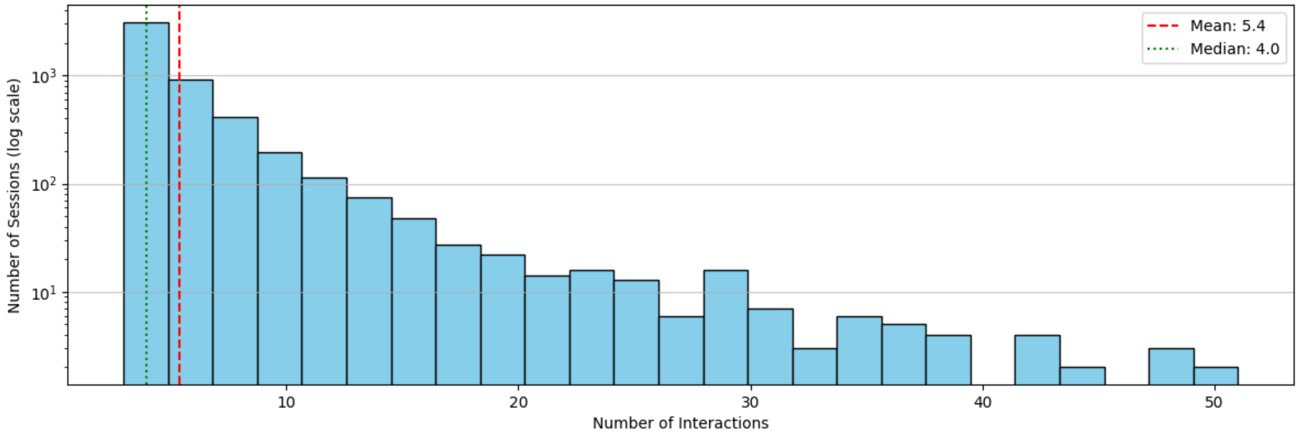


Figure 2 - Distribution of interactions per session

4.4.2 Item Popularity Distribution

The distribution of job applications across job postings was highly unbalanced. While no job received more than 14 applications, the vast majority were applied to only once. Specifically, the number of interactions per job ranged from 1 to 14, with a mean of 1.3. The 25th, 50th (median), and 75th percentiles were all 1, underscoring the sparsity and flatness of the interaction distribution. Specifically, the number of interactions per job ranged from 1 to 14, with a mean of 1.3. The 25th, 50th (median), and 75th percentiles were all 1, indicating that the vast majority of jobs were applied to only once. This "long-tail" pattern means that popular items dominate the dataset, but many jobs are only applied to a few times. This observation highlights the importance of including novelty-focused evaluation metrics like EPC, and of designing recommendation strategies that avoid always promoting the most popular items.

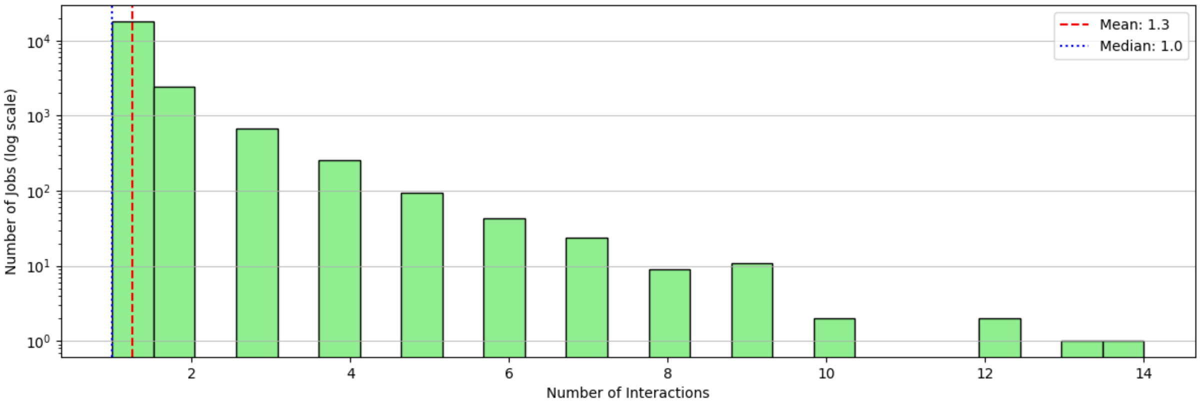


Figure 3 - Distribution of interactions per job (item_id)

Table 2 - Basic statistics for interactions per job (item_id)

Statistic	Value
Mean	1.3
Max	14
Min	1
Q1 (25%)	1.0
Median (50%)	1.0
Q3 (75%)	1.0
Count	21532

4.4.3 State Feature Distribution

The job postings covered all 50 U.S. states, Washington D.C., and several territories, with highly imbalanced representation. The top five states, Texas (2,513), Florida (2,256), California (1,969), Illinois (1,572), and New York (1,292), accounted for a large share of postings. Conversely, states like Maine and Wyoming appeared only once, and several others had counts in the single digits. This significant disparity reflects regional biases in the original dataset and underlines the necessity of one-hot encoding for categorical state features, ensuring that rare states are not lost or misrepresented during vectorization.

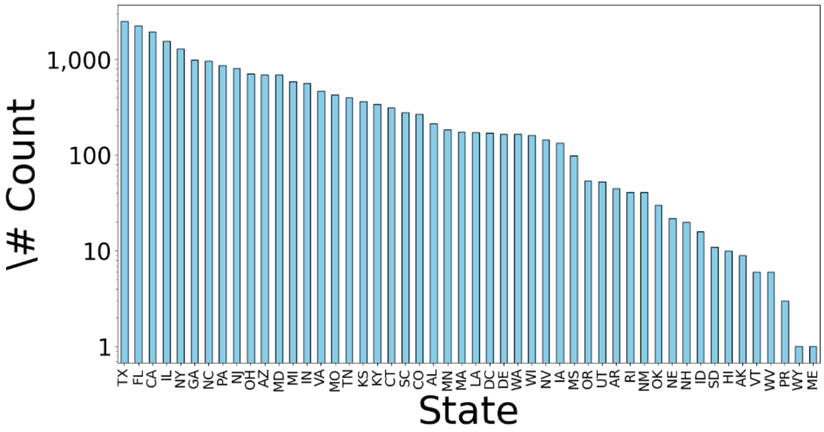


Figure 4 - Distribution of states

4.4.4 BERT Embedding

The job content was embedded using all-MiniLM-L6-v2, producing 384-dimensional vectors per text field. Across 21,532 job postings, there were 21,332 unique embeddings, yielding a high uniqueness ratio of approximately 0.991. This suggests that the embedding process captured fine-grained semantic variation across job descriptions. The value range observed in the embedding columns was approximately -0.278 to 0.406, consistent with expected bounds for this BERT model.

Chapter 5 - Results and Discussion

Content:

This section presents the performance of the three autoencoder-based recommender models, AE, DAE, and VAE, on the CareerBuilder 2012 dataset. Their effectiveness was evaluated across multiple metrics under both next-item and remaining-items prediction settings.

5.1 Evaluation by Metric and Scenario

5.1.1 Accuracy Metrics

Each model was evaluated on accuracy (nDCG@k, MRR@k) and beyond-accuracy (EPC, EPD, Item Coverage@k, Session Coverage@k) metrics. Results were computed for both next-item prediction and remaining-items-in-session prediction.

Quantitatively, the models exhibited a wide but informative performance distribution. For next-item prediction, nDCG scores ranged from 0.0113 to 0.0409, with a mean of 0.0324, while MRR ranged from 0.0113 to 0.0293 (mean 0.0246). In the remaining-items setting, nDCG varied from 0.0653 to 0.2778 (mean 0.2086) and MRR from 0.1882 to 0.2615 (mean 0.2188). These results confirm that while absolute accuracy is moderate, the models perform better when given a broader session context.

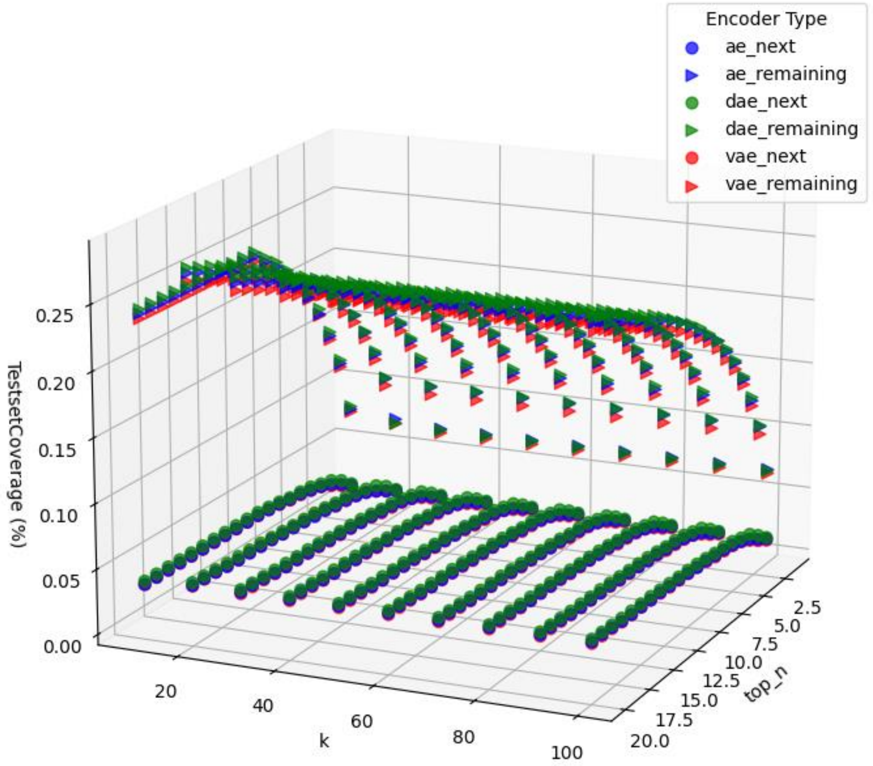


Figure 5 - nDCG performance across different recommendation list sizes (k) and top-n recommended job thresholds for each autoencoder architecture (AE, DAE, VAE), evaluated on two prediction tasks: next-item prediction and remaining-items prediction.

5.1.2 Beyond-Accuracy Metrics

EPC scores were modest but consistent, averaging 0.0008 for next-item and 0.0167 for remaining-items predictions, indicating that the models did recommend some less frequent jobs, though overall novelty was low. EPD values were extremely close to zero in both cases, highlighting the homogeneity of recommendations and a likely inability to break semantic redundancy within sessions.

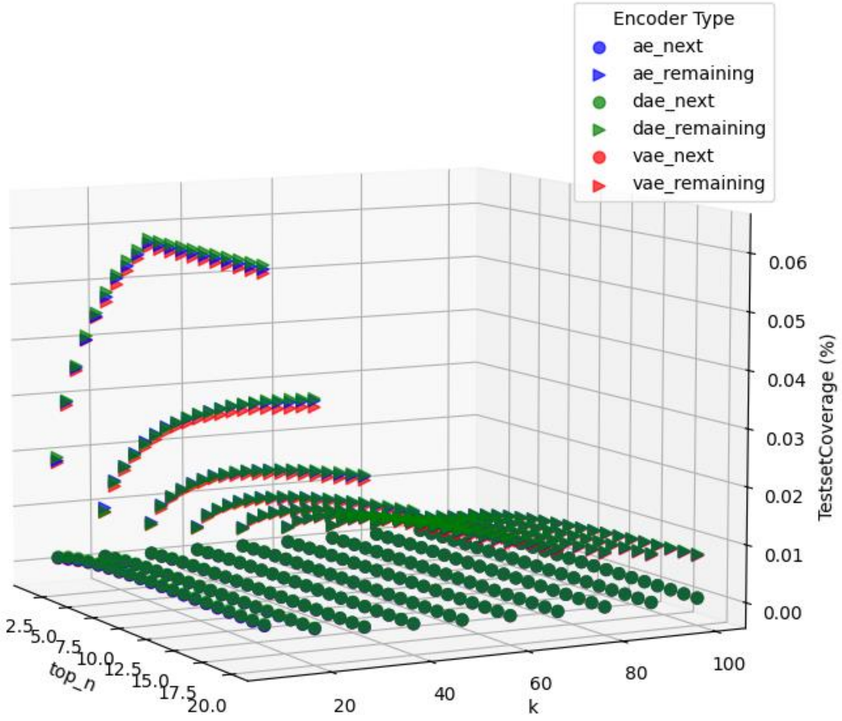


Figure 6 - EPC performance across different recommendation list sizes (k) and top-n recommended job thresholds for each autoencoder architecture (AE, DAE, VAE), evaluated on two prediction tasks: next-item prediction and remaining-items prediction.

Coverage metrics further contextualize these results. Catalog coverage ranged from 1.29% to 13.73%, with a mean around 8.10%, reflecting moderate item space diversity. Test set (session) coverage ranged from 13.41% to 75.18% (mean 55.42%), indicating that the models could serve over half of the sessions but struggled with edge cases or highly sparse histories.

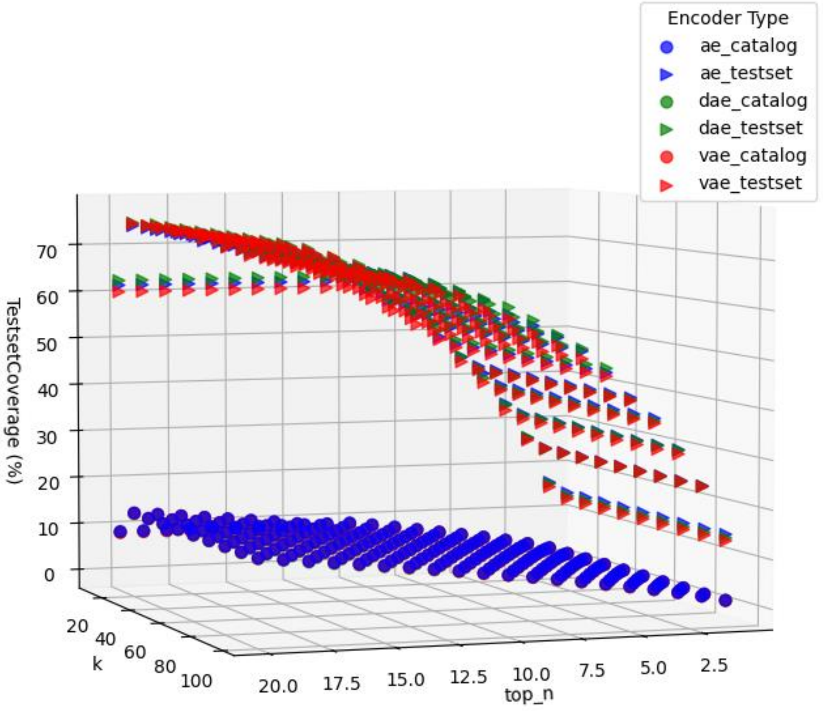


Figure 7 - Coverage performance across different recommendation list sizes (k) and top-n recommended job thresholds for each autoencoder architecture (AE, DAE, VAE), evaluated on the whole catalog of jobs and on the testset set of jobs.

5.1.3 Performance by Task

When examining the individual BERT-enhanced models, AE_NLP, DAE_NLP, and VAE_NLP displayed differentiated strengths depending on the evaluation setting:

- In the next-item prediction task, DAE_NLP slightly outperformed AE_NLP and VAE_NLP in terms of nDCG and MRR. This suggests that the DAE architecture, despite generalization issues, is able to capture immediate sequential signals better when sessions are short and interaction targets are local. AE_NLP was consistent but slightly less responsive to short-term dynamics. VAE_NLP underperformed in raw nDCG@k on this task but remained stable across variations in k.
- In the remaining-items prediction setting, AE_NLP led the models in nDCG and MRR, suggesting that it generalizes better across longer interaction windows and provides strong sequential coherence. DAE_NLP followed closely but showed a slight drop in robustness, while VAE_NLP demonstrated the smoothest metric evolution across different top_n values. Although its accuracy was slightly lower, VAE_NLP showed the best potential in terms of maintaining semantic diversity and avoiding overfitting to popular items.

These model-level insights confirm that AE_NLP is a strong baseline for general session modeling, DAE_NLP is best tuned for immediate prediction tasks, and VAE_NLP, while not leading in raw accuracy, delivers more stable and coherent performance when diversity, novelty, and semantic structure are prioritized.

Overall, while performance is generally stable and consistent, recommendation quality is still constrained by session sparsity and item redundancy. Nonetheless, VAE models with BERT embeddings show strong generalization and robustness, especially in broader prediction tasks, making them promising for practical use and further enhancement via hybrid modeling or re-ranking techniques.

5.2 Comparison with Lacic et al. (2020)

A direct comparison between this study and Lacic et al. (2020) reveals both overlap and divergence in performance. As shown in its Figure 8, the models from this study (marked *_nlp) consistently outperform Lacic's in terms of nDCG@20 across all values of k. For instance, the best Lacic model (AE_Comb_Lacic) plateaus around 0.124, while the BERT-based variants in this study (e.g., dae_comb_nlp and ae_comb_nlp) reach nDCG values above 0.275, more than doubling the top scores reported in the original paper.

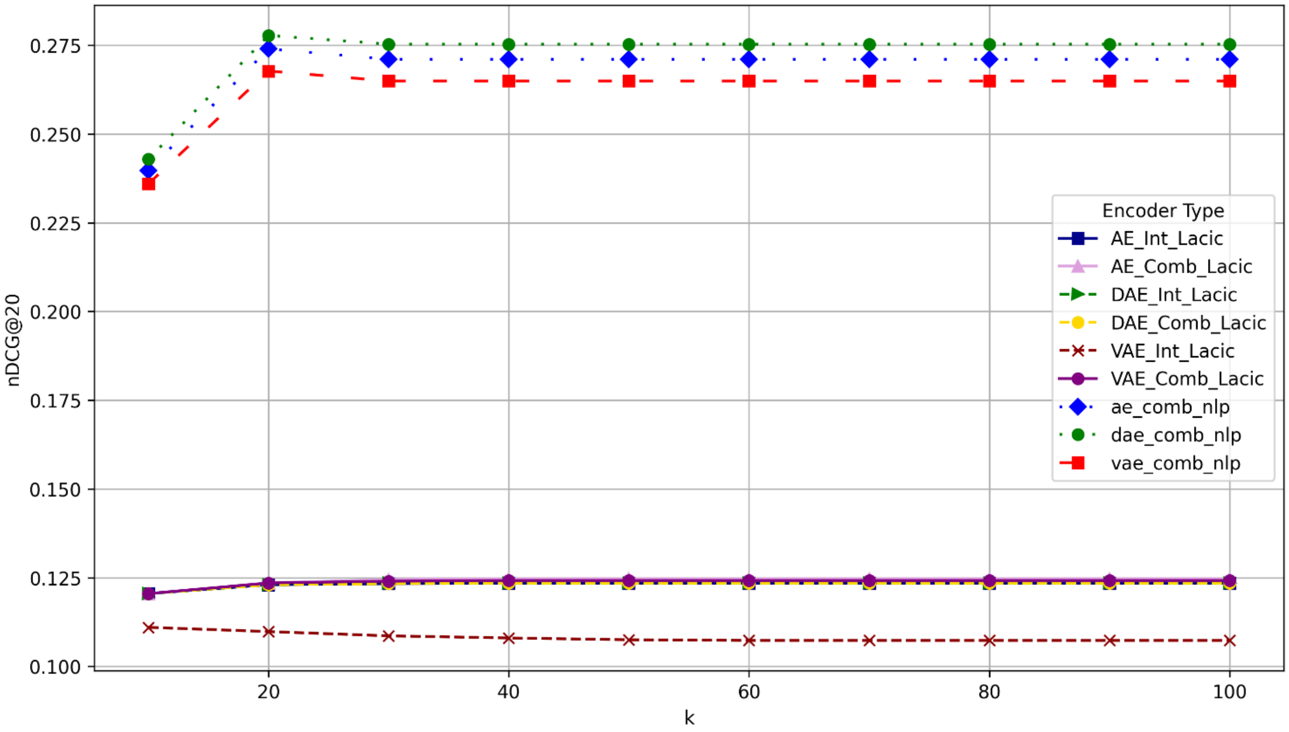


Figure 8 - nDCG vs. k similar sessions for top n recommended jobs equals 20.

5.2.1 Accuracy Gains

This performance gain likely stems from the use of BERT embeddings in place of hand-engineered topic distributions. Semantic encoders like BERT enable more expressive content vectors, improving alignment between job postings and session context, especially when item overlap is sparse. The strong results also reflect the scalability of the NLP-enhanced approach: once embedded, job vectors capture contextual cues that topic models cannot easily encode.

Interestingly, while VAE underperformed Lacic's AE and DAE in their setup (VAE_Int_Lacic ~0.108), BERT-based variant (vae_comb_nlp) stabilized around 0.265 in this study. Although it trailed AE and DAE in raw nDCG here, its consistency across k suggests better semantic coherence and robustness, validating the theoretical claim that VAE is more expressive in latent modeling when supplied with rich inputs.
This interpretation is supported by Figure 9, which compares the next-item nDCG@k scores of various baselines and the BERT-enhanced autoencoders. Although Lacic's traditional models such as GRU4Rec, iKNN, and VsKNN dominate in top-n recommendation performance, dae_comb_nlp and ae_comb_nlp from this study exhibit stable but lower nDCG curves. Notably, vae_comb_nlp performs worst among the three, underscoring that while VAE is robust in full-session modeling, it is less suited for immediate next-item prediction without additional refinement. This suggests that DAE's slight advantage in short-range tasks arises from its encoding sensitivity, while VAE's strength lies in semantic continuity over longer prediction horizons.

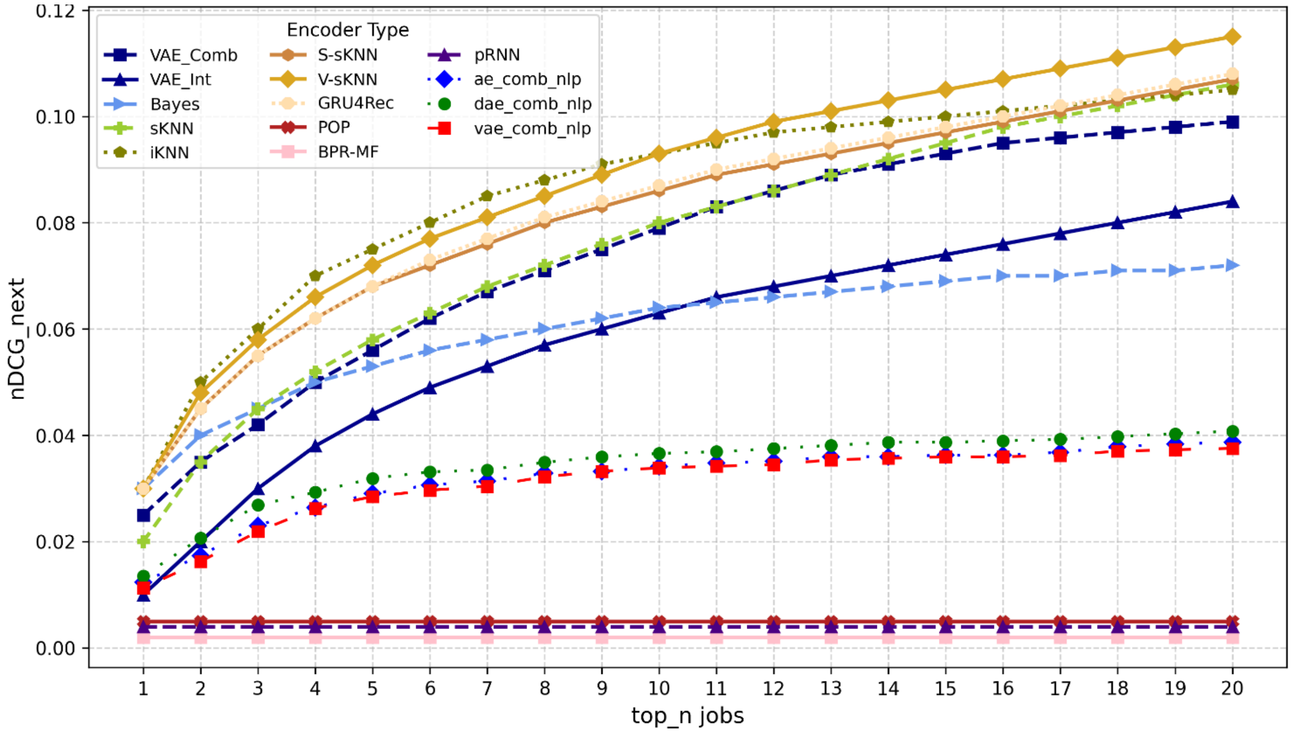


Figure 9 - nDCG vs. top n recommended jobs for k similar sessions equals 60.

5.2.2 Novelty Trade-offs

This interpretation is further supported by Figure 10, which compares EPC scores across top-n values. Here, vae_comb_nlp, despite its weaker nDCG performance, shows marginally better EPC values than the other two NLP models. This aligns with Lacic et al.'s findings that VAE variants, particularly VAE_Comb, tend to outperform others in novelty-focused metrics. The consistent EPC trend of vae_comb_nlp suggests that its latent space captures less popular items more effectively, even if they are not ranked highly. Meanwhile, dae_comb_nlp and ae_comb_nlp achieve lower EPC values, showing that their predictions are more skewed toward popular items.

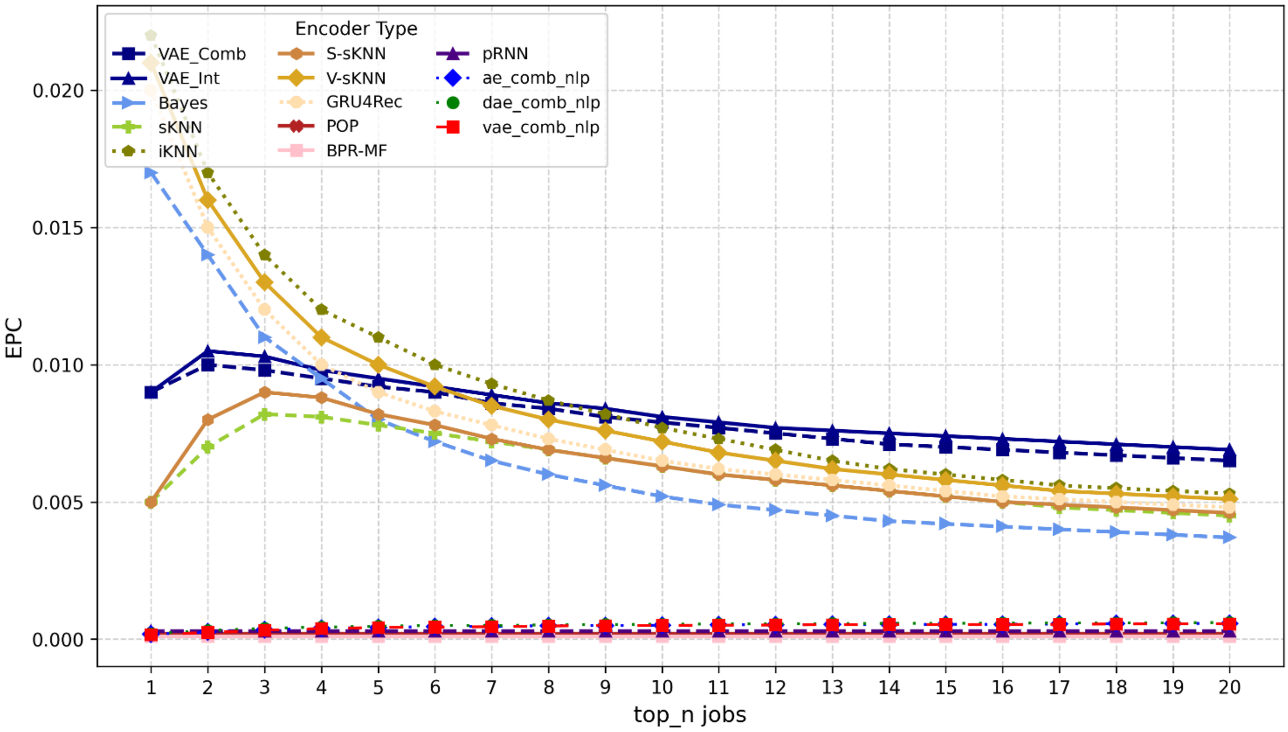
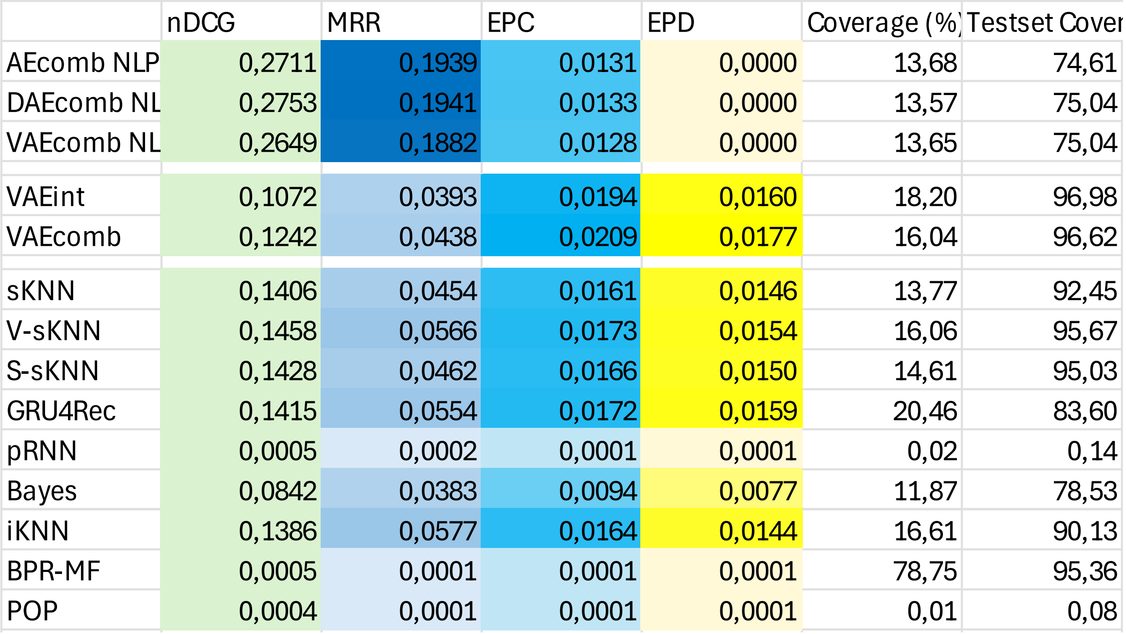


Figure 10 - EPC vs. top n recommended jobs for k similar sessions equals 60.

5.2.3 Strengths and Limitations

Table 3 from the extended benchmark (k = 20) confirms these observations under the remaining-items prediction setting. While AEcomb_NLP and DAEcomb_NLP achieve the highest nDCG scores (0.2711 and 0.2753, respectively), their novelty metrics (EPC ≈ 0.0131–0.0133 and EPD = 0.0000) are flat, suggesting a strong bias toward frequently seen items. In contrast, Lacic's VAE_Comb model achieved lower accuracy (nDCG = 0.1242) but substantially higher novelty (EPC = 0.0209, EPD = 0.0177). These results reinforce Lacic's conclusion that novelty is better optimized through variational latent models than purely feedforward or denoising ones.

Table 3 - Prediction results (k = 20) of remaining jobs that will be subject to interaction within a session.



	nDCG	MRR	EPC	EPD	Coverage (%)	Testset Cover
AEcomb NLP	0,2711	0,1939	0,0131	0,0000	13,68	74,61
DAEcomb NL	0,2753	0,1941	0,0133	0,0000	13,57	75,04
VAEcomb NL	0,2649	0,1882	0,0128	0,0000	13,65	75,04
VAEint	0,1072	0,0393	0,0194	0,0160	18,20	96,98
VAEcomb	0,1242	0,0438	0,0209	0,0177	16,04	96,62
sKNN	0,1406	0,0454	0,0161	0,0146	13,77	92,45
V-sKNN	0,1458	0,0566	0,0173	0,0154	16,06	95,67
S-sKNN	0,1428	0,0462	0,0166	0,0150	14,61	95,03
GRU4Rec	0,1415	0,0554	0,0172	0,0159	20,46	83,60
pRNN	0,0005	0,0002	0,0001	0,0001	0,02	0,14
Bayes	0,0842	0,0383	0,0094	0,0077	11,87	78,53
iKNN	0,1386	0,0577	0,0164	0,0144	16,61	90,13
BPR-MF	0,0005	0,0001	0,0001	0,0001	78,75	95,36
POP	0,0004	0,0001	0,0001	0,0001	0,01	0,08

However, the VAEcomb_NLP model in this study did not replicate the novelty boost seen in Lacic's VAE_Comb. Despite using richer BERT content, its EPC (0.0128) and EPD (0.0000) remained low, likely due to semantic overlap in session history and short session lengths limiting novelty space. Still, all three BERT-based models achieved significantly higher accuracy and competitive coverage relative to classic baselines like sKNN or GRU4Rec. With over 74% test set coverage and ~13.6% catalog coverage, the NLP-enhanced models scale well while maintaining high recommendation relevance.

5.2.4 Summary of Findings

In summary, BERT embeddings clearly enhance accuracy, while the role of VAE in fostering diversity depends heavily on task framing (next vs. remaining items), content variance, and list depth. Future work might focus on augmenting VAE-based recommenders with diversity-aware objectives or post-hoc re-ranking to further boost novelty without harming relevance.

Overall, while performance is generally stable and consistent, recommendation quality is still constrained by session sparsity and item redundancy. Nonetheless, VAE models with BERT embeddings show strong generalization and robustness, especially in broader prediction tasks, making them promising for practical use and further enhancement via hybrid modeling or re-ranking techniques.

In line with Lacic's insights, novelty is harder to optimize than accuracy and typically requires models with probabilistic representations or explicit mechanisms for diversity. The BERT-VAE combination, though modest in ranking accuracy, appears to naturally balance popularity bias and semantic drift, enabling a broader exposure to long-tail items.

Altogether, this comparison supports the idea that modern language models not only simplify preprocessing but also significantly enhance ranking quality in session-based recommendation.

5.3 Hypotheses Revisited

This subsection evaluates the four core research hypotheses defined in Section 3.4.1 in light of the experimental results presented above:

- H1 (Semantic Encoding Improvement): Autoencoders trained on BERT-based embeddings were expected to improve semantic novelty and diversity, as measured by EPC and EPD. However, this hypothesis was not confirmed. Despite using richer and more expressive input vectors than Lacic's topic-model-based features, EPC and EPD scores remained very low across all BERT-enhanced models. In the remaining-items task, all three variants (AE, DAE, VAE) achieved EPD = 0.0000, indicating no meaningful increase in profile diversity. The EPC values also fell short of Lacic's VAE_Comb baseline. This suggests that BERT embeddings alone do not guarantee novelty unless explicitly optimized for it.

- H2 (Accuracy Preservation): This hypothesis is confirmed. All BERT-based models in this study achieved nDCG@20 scores of ~0.27 for remaining-items prediction, more than double the accuracy reported for Lacic's best models (≈0.12). This validates that replacing sparse topic vectors with dense BERT embeddings not only preserves but significantly enhances ranking performance, particularly for full-session modeling.

- H3 (VAE Expressiveness): The hypothesis that VAE would produce the most expressive and semantically rich recommendations, reflected by the highest EPC and EPD scores, is partially confirmed. In next-item prediction (Figure 9), VAE showed marginally better EPC trends than AE and DAE, which is consistent with Lacic et al.'s findings. However, in the remaining-items prediction (Table 4), VAEcomb NLP did not surpass its AE and DAE counterparts in novelty or accuracy. Therefore, while the model structure is well-suited for expressiveness, its advantage did not consistently emerge without additional regularization or diversity constraints.

- H4 (Robustness of Denoising Autoencoders): This hypothesis is not confirmed. Although DAEcomb NLP achieved the highest nDCG and MRR in both prediction tasks, its advantage was marginal and not clearly associated with greater robustness to noise or sparsity. Its novelty and coverage scores remained nearly identical to AEcomb NLP, and performance across session types showed no distinctive benefit from noise injection. This suggests that denoising is less effective when semantic embeddings are already continuous and smooth.

Chapter 6 - Conclusion

Content:

6.1 Summary of Findings

This thesis explored how session-based job recommender systems can be improved by integrating deep learning architectures, semantic content embeddings, and multimodal data fusion. Building on the foundational work by Lacic et al. (2020), who demonstrated the effectiveness of autoencoder-based representations for session modeling, this study introduced BERT-based content embeddings to replace the original one-hot topic encodings. The resulting hybrid representation combines structured and semantic information, leading to a richer and more expressive session encoding.

Using the CareerBuilder 2012 dataset, we trained and evaluated three types of autoencoders: AE, DAE, and VAE. Each model was tested on two evaluation tasks, predicting the next job in a session and predicting all remaining jobs. Recommendation quality was assessed using accuracy (nDCG, MRR) and beyond-accuracy metrics (EPC, EPD, item coverage, and session coverage).

6.2 Key Findings

The results validate several hypotheses. First, the substitution of BERT embeddings for one-hot topic encodings significantly improved accuracy across all autoencoder variants, doubling the nDCG scores reported in the original study. This confirms that richer semantic content improves session representation quality. Second, while variational autoencoders (VAE) did not consistently outperform other models in terms of raw ranking accuracy, they showed greater consistency across evaluation settings and stronger potential for novelty-focused recommendation, aligning with Lacic et al.'s observations.

6.3 Limitations

However, the study also uncovered limitations. Contrary to expectations, the use of BERT embeddings did not significantly improve beyond-accuracy metrics such as EPC or EPD. This suggests that while semantic content enhances accuracy, it does not guarantee novelty or diversity without explicit optimization for these goals. Similarly, the denoising autoencoder (DAE) showed only marginal gains in robustness, highlighting that input noise may be less effective when dealing with already dense and informative embeddings.

6.4 Future Research Directions

Overall, this work confirms that deep semantic embeddings can significantly enhance the performance of session-based recommender systems and provides empirical evidence that variational models retain value for applications prioritizing diversity and robustness. Future work should focus on combining these models with re-ranking techniques or diversity-aware objectives to fully realize the potential of novelty-aware job recommendations. Furthermore, integrating multimodal content and exploring transformer-based architectures may yield additional gains, especially under real-world constraints of session anonymity, content sparsity, and user cold starts.

References

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer.

Ailyn, D. (2024). Multimodal Data Fusion Techniques. ResearchGate Preprint. Retrieved from https://www.researchgate.net/publication/383887675

Al-Otaibi, S. T., & Ykhlef, M. (2012). A survey of job recommender systems. International Journal of the Physical Sciences, 7(29), 5127–5142. https://doi.org/10.5897/IJPS12.482

Analytics Vidhya. (2024). Understanding Hit Rate, MRR, and MMR Metrics. Retrieved from https://www.analyticsvidhya.com

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(2), 423-443.

Bennouna, K., Chougrad, H., Khamlichi, Y., El Boushaki, A., & El Haj Ben Ali, S. (2022). Variational Autoencoders Versus Denoising Autoencoders for Recommendations. In WITS 2020, Lecture Notes in Electrical Engineering, 745, 179–187.

Bhatia, R. (2023). Recommendation System Evaluation Metrics. Medium. Retrieved from https://medium.com/@rishabhbhatia315

Binns, R., Veale, M., Van Kleek, M., & Shadbolt, N. (2018). 'It's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 1–14.

Bobadilla, J., Ortega, F., Hernando, A., & Gutierrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109–132.

Borisyuk, F., Zhang, L., & Kenthapadi, K. (2017). LiJAR: A system for job application redistribution towards efficient career marketplace. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1397–1406.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331–370.

Chen, L., Ma, R., Hannák, A., & Wilson, C. (2018). Investigating the impact of gender on rank in resume search engines. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 1–14.

Cherifi, I. (2023). Session-based recommender systems. LinkedIn. https://www.linkedin.com/pulse/session-based-recommender-systems-imane-cherifi

Conceição, F. L. A., Pádua, F. L. C., Lacerda, A., Machado, A. C., & Dalip, D. H. (2019). Multimodal data fusion framework based on autoencoders for top-N recommender systems. Applied Intelligence, 49(8), 3267–3283. https://doi.org/10.1007/s10489-019-01430-7

de Ruijt, C., & Bhulai, S. (2021). Job recommender systems: A review. arXiv preprint arXiv:2111.13576.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, 4171–4186.

Evidently AI. (2023). Evaluating Recommender Systems. Retrieved from https://www.evidentlyai.com/ranking-metrics/evaluating-recommender-systems

Ferreira, D., Silva, S., Abelha, A., & Machado, J. (2020). Recommendation System Using Autoencoders. Applied Sciences, 10(16), 5510. https://doi.org/10.3390/app10165510

Gaw, S., Rao, R., Patel, R., & McLaughlin, S. (2022). Multimodal data fusion for system improvement: A review. AFIT Technical Report. Retrieved from https://www.afit.edu/BIOS/publications/Gawetal.2022_MultimodaldatafusionforsystemsimprovementAreview.pdf

Gedikli, F., & Jannach, D. (2026). Semantics-based recommender systems. In Encyclopedia of Social Network Analysis and Mining (3rd ed.). Springer.

Geyik, S. C., Ambler, S., & Kenthapadi, K. (2019). Fairness-aware ranking in search & recommendation systems with application to LinkedIn Talent Search. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2221–2231.

Hidasi, B., et al. (2023). Widespread Flaws in Offline Evaluation of Recommender Systems. Retrieved from https://hidasi.eu/assets/pdf/eval_flaws_recsys23.pdf

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In Proceedings of the International Conference on Learning Representations (ICLR).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647

Holstein, K., Wortman Vaughan, J., Daumé, H., Dudik, M., & Wallach, H. (2019). Improving fairness in machine learning systems: What do industry practitioners need? Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 1–16.

Hu, J., Xia, W., Zhang, X., Fu, C., Wu, W., Huan, Z., Li, A., Tang, Z., & Zhou, J. (2024). Enhancing sequential recommendation via LLM-based semantic embedding learning. In Companion Proceedings of the ACM Web Conference 2024 (WWW '24). https://doi.org/10.1145/3589335.3648307

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (pp. 263–272).

Jannach, D. (2018). Session-based recommendation: Challenges and recent advances. In IIR 2018 - Italian Information Retrieval Workshop.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. 37(2), 183–233 (1999)

Khan, U. A. (2023). Text embedding and sentence similarity retrieval at scale with Amazon SageMaker. https://aws.amazon.com/blogs/machine-learning/text-embedding-and-sentence-similarity-retrieval-at-scale-with-amazon-sagemaker-jumpstart/

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114. https://arxiv.org/abs/1312.6114

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2), 233-243.

Lacic, E., Reiter-Haas, M., Kowald, D., Dareddy, M. R., Cho, J., & Lex, E. (2020). Using autoencoders for session-based job recommendations. User Modeling and User-Adapted Interaction, 30, 617–658. https://doi.org/10.1007/s11257-020-09269-1

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (pp. 1188–1196).

Liang, D., Krishnan, R. G., Hoffman, M. D., & Jebara, T. (2018). Variational Autoencoders for Collaborative Filtering. Proceedings of the 2018 World Wide Web Conference, 689–698. https://doi.org/10.1145/3178876.3186150

Liang, S., Pan, Z., Liu, W., Yin, J., & de Rijke, M. (2024). A Survey on Variational Autoencoders in Recommender Systems. ACM Computing Surveys. https://doi.org/10.1145/3663364

Liu, D., Cheng, H., Chen, M., Liu, Y., & Wang, Y. (2020). Semantic content-based recommendation of software services. Journal of Systems and Software, 170, 110740. https://doi.org/10.1016/j.jss.2020.110740

Liu, J., Zhu, Y., & Xie, R. (2024). Effective techniques for multimodal data fusion: A comparative analysis. Information Fusion, 99, 104021.

Liu, Q., Hu, J., Xiao, Y., Zhao, X., Gao, J., Wang, W., Li, Q., & Tang, J. (2024). Multimodal recommender systems: A survey. arXiv preprint arXiv:2302.03883. https://arxiv.org/abs/2302.03883

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1831–1839). https://doi.org/10.1145/3219819.3220023

Ludewig, M., & Jannach, D. (2018). Evaluation of session-based recommendation algorithms. arXiv preprint arXiv:1803.09587.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... & Lee, S.-I. (2021). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67.

Mashayekhi, Y., Li, N., Kang, B., Lijffijt, J., & De Bie, T. (2023). A challenge-based survey of e-recruitment recommendation systems. arXiv preprint arXiv:2209.05112v2.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273–1282.

Messica, A., Rokach, L., & Friedman, M. (2017). Session-based recommendations using item embedding. In Proceedings of the 22nd International Conference on Intelligent User Interfaces (pp. 377–381). ACM. https://doi.org/10.1145/3025171.3025203

Mhamdi, D., Moulouki, R., El Ghoumari, M. Y., Azzouazi, M., & Moussaid, L. (2020). Job recommendation based on job profile clustering and job seeker behavior. Procedia Computer Science, 175, 695–699. https://doi.org/10.1016/j.procs.2020.07.102

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

MobiDev. (2024). Choosing the best approach to building an NLP-based text recommendation system. Retrieved from https://mobidev.biz/blog/natural-language-processing-nlp-use-cases-business

Moreira, G. de S. P., Rabhi, S., Ak, R., Kabir, M. Y., & Oldridge, E. (2021). Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. Proceedings of the ACM SIGIR Workshop on eCommerce (SIGIR eCom'21). https://arxiv.org/abs/2110.01001

Moreira, G., et al. (2021). Transformers4Rec: Bridging the gap between NLP and sequential/session-based recommendation. arXiv preprint arXiv:2104.09938.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. Proceedings of the 28th International Conference on Machine Learning (ICML-11), 689–696.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325–341). Springer.

Readsumant. (2023). Understanding NDCG as a Metric for Your Recommendation System. Medium. Retrieved from https://medium.com/@readsumant/understanding-ndcg-as-a-metric-for-your-recomendation-system-5cd012fb3397

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. EMNLP-IJCNLP 2019, 3982–3992.

Reusens, M., Lemahieu, W., Baesens, B., & Sels, L. (2018). Evaluating recommendation and search in the labor market. Knowledge-Based Systems, 152, 62–69.

Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). AutoRec: Autoencoders Meet Collaborative Filtering. Proceedings of the 24th International Conference on World Wide Web, 111–112. https://doi.org/10.1145/2740908.2742726

Sonboli, N., Wang, Y., Burke, R., & Mobasher, B. (2021). Fairness-aware recommendation with adaptive pairwise reweighting. Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, 169–177.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 1441–1450. https://doi.org/10.1145/3357384.3357895

Vančura, V., Kordík, P., & Straka, M. (2024). BeeFormer: Bridging the gap between semantic and interaction similarity in recommender systems. In Proceedings of the 18th ACM Conference on Recommender Systems (RecSys '24). https://doi.org/10.1145/3640457.3691707

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and Composing Robust Features with Denoising Autoencoders. Proceedings of the 25th International Conference on Machine Learning, 1096–1103. https://doi.org/10.1145/1390156.1390294

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Improving text embeddings with large language models. ACL 2024, 11897–11916.

Wang, Q., Zhang, X., & Yu, H. (2023). A survey on session-based multimodal recommender systems. ACM Transactions on Recommender Systems, 11(3), 1-30.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR), 54(7), 1–38.

Wang, Y., Zhang, H., Liu, Z., Yang, L., & Yu, P. S. (2022c). ContrastVAE: Contrastive variational autoencoder for sequential recommendation. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2056–2066.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 346–353.

Yan, E. (2023). Using large language models as recommendation systems. Towards Data Science. https://towardsdatascience.com/using-large-language-models-as-recommendation-systems-49e8aeeff29b/

Yan, Ziyou. (Mar 2025). Improving Recommendation Systems & Search in the Age of LLMs. eugeneyan.com. https://eugeneyan.com/writing/recsys-llm/

Yang, S., Huang, C., & Zhu, Y. (2024). Enhancing sequential recommendation via LLM-based semantic matching. arXiv preprint arXiv:2409.10309.

Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2020). A hybrid co-attention network for multi-modal recommendation. IEEE Transactions on Knowledge and Data Engineering, 34(3), 1122–1135.

Zhang, G., Liu, Y., & Jin, X. (2020). A Survey of Autoencoder-Based Recommender Systems. Frontiers of Computer Science, 14(2), 430–450. https://doi.org/10.1007/s11704-018-8052-6

Zhang, L., Chen, J., & Zhao, W. (2022). Late fusion for multimodal recommender systems. Expert Systems with Applications, 184, 115522.

Zhang, T., Sun, Y., & Wang, X. (2024). Rethinking LLM architectures for recommendation systems. KDnuggets. https://www.kdnuggets.com/latest-innovations-in-recommendation-systems-with-llms

Zhang, W., Zhang, M., & Ma, S. (2023). Beyond clicks: Multimodal user modeling for session-based recommendation. Frontiers in Big Data, 6, 1295009.

Zhao, F., Zhang, C., & Geng, B. (2024). Deep multimodal data fusion. ACM Computing Surveys, 56(9), Article 216. https://doi.org/10.1145/3649447

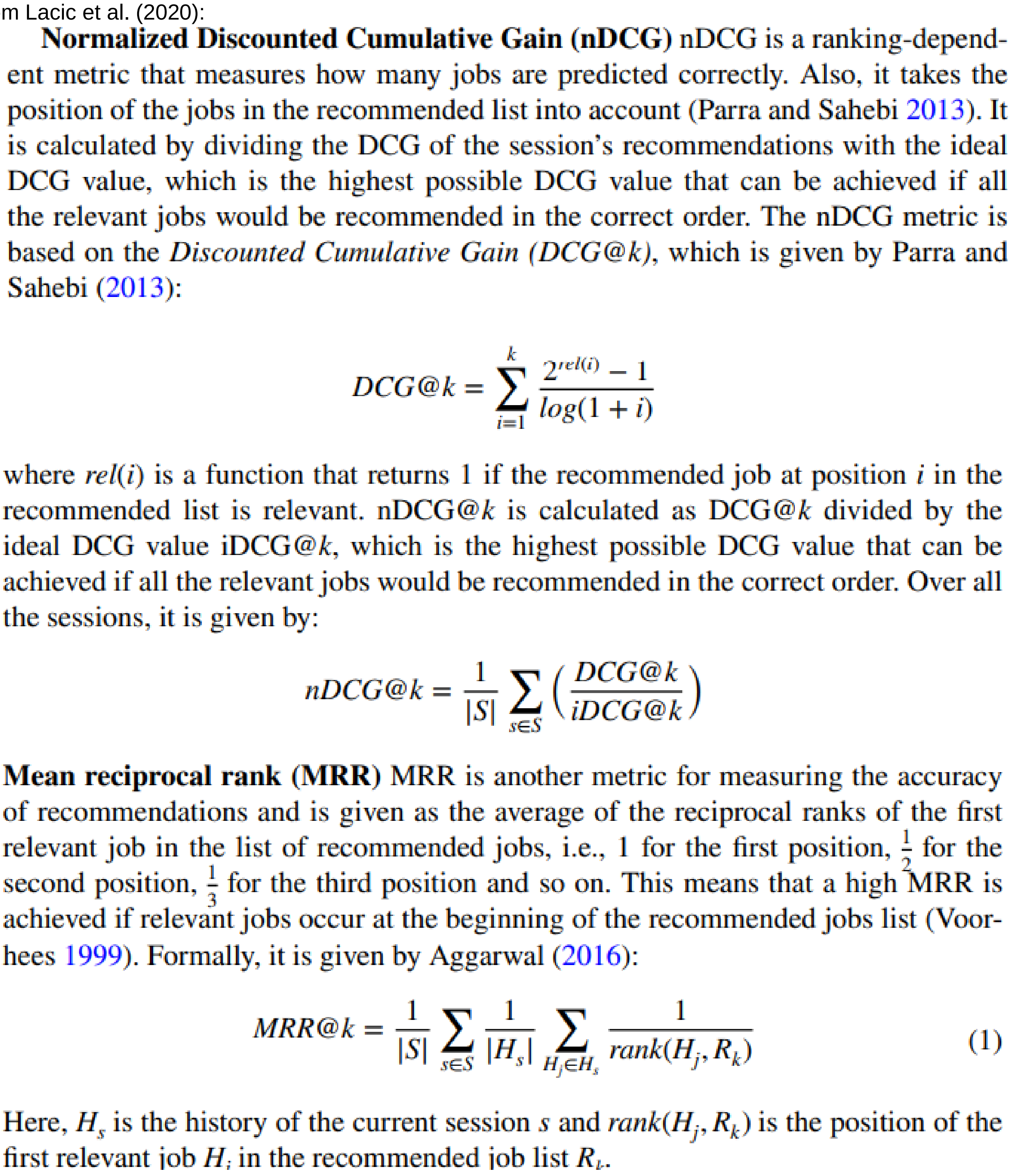
Zhao, L., Chen, H., Liu, B., Lin, Q., & Hu, W. (2023). Two-sided fairness in job recommender systems. arXiv preprint arXiv:2304.05428.

Zhao, Z., Yang, Z., & Li, X. (2023). Modeling micro-behaviors for hyper-personalized job recommendation. Knowledge-Based Systems, 269, 110239.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving Recommendation Lists Through Topic Diversification. In Proceedings of the 14th International Conference on World Wide Web (WWW '05), 22–32.

APPENDIX

From Lacic et al. (2020):



Normalized Discounted Cumulative Gain (nDCG) nDCG is a ranking-dependent metric that measures how many jobs are predicted correctly. Also, it takes the position of the jobs in the recommended list into account (Parra and Sahebi 2013). It is calculated by dividing the DCG of the session's recommendations with the ideal DCG value, which is the highest possible DCG value that can be achieved if all the relevant jobs would be recommended in the correct order. The nDCG metric is based on the *Discounted Cumulative Gain (DCG@k)*, which is given by Parra and Sahebi (2013):

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{log(1 + i)}$$

where $rel(i)$ is a function that returns 1 if the recommended job at position i in the recommended list is relevant. nDCG@k is calculated as DCG@k divided by the ideal DCG value iDCG@k, which is the highest possible DCG value that can be achieved if all the relevant jobs would be recommended in the correct order. Over all the sessions, it is given by:

$$nDCG@k = \frac{1}{|S|} \sum_{s \in S} \left(\frac{DCG@k}{iDCG@k} \right)$$

Mean reciprocal rank (MRR) MRR is another metric for measuring the accuracy of recommendations and is given as the average of the reciprocal ranks of the first relevant job in the list of recommended jobs, i.e., 1 for the first position, $\frac{1}{2}$ for the second position, $\frac{1}{3}$ for the third position and so on. This means that a high MRR is achieved if relevant jobs occur at the beginning of the recommended jobs list (Voorhees 1999). Formally, it is given by Aggarwal (2016):

$$MRR@k = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|H_s|} \sum_{H_j \in H_s} \frac{1}{rank(H_j, R_k)} \quad (1)$$

Here, H_s is the history of the current session s and $rank(H_j, R_k)$ is the position of the first relevant job H_j in the recommended job list R_k.

System-based novelty (EPC) System-based novelty denotes the ability of a recommender to introduce sessions to job postings that have not been (frequently) experienced before in the system. A recommendation that is accurate but not novel will include items that the session user enjoys, but (probably) already knows. Optimizing system-based novelty has been shown to have a positive, trust-building impact on user satisfaction (Pu et al. 2011). Moreover, system-based novelty is also an important metric for the job domain since applying to popular jobs may decrease a user's satisfaction due to high competition and less chance of getting hired (see, e.g., Kenthapadi et al. 2017). In our experiments, we measure the system-based novelty using the expected popularity complement (EPC) metric introduced by Vargas and Castells (2011). In contrast to solely popularity-based metrics (e.g., Zhou et al. 2010), EPC also accounts for the recommendation rank and the relevance for the current session. Thus, the system-based novelty $nov_{system}(R_k|s)$ for the recommendation list R_k of length k for session s is given by:

$$EPC@k = \frac{1}{|S|}\sum_{s\in S}\frac{1}{|R_k|}\sum_{R_i\in R_k} disc(i)p(rel|R_i, s)(1 - p(seen|R_i))$$

Here, $disc(i)$ is a discount factor to weight the recommendation rank i [i.e., $disc(i) = 1/log_2(i + 1)$] and $p(rel|R_i, s)$ is 1 if the recommended job R_i is relevant for session s or 0 otherwise (i.e., only jobs that are in the current session history are taken into account). Finally, $p(seen|R_i)$ defines the probability that a recommended job R_i was already seen in the system, i.e., $p(seen|R_i) = log_2(pop_{R_i} + 1)/log_2(pop_{MAX} + 1)$.

Session-based novelty (EPD) In contrast to system-based novelty, session-based novelty incorporates the semantic content of jobs and represents how *surprising* or *unexpected* the recommendations are for a specific session history (Zhang et al. 2012). Given a distance function $d(H_i, H_j)$ that represents the dissimilarity between two jobs H_i and H_j, the session-based novelty is given as the average dissimilarity of all job pairs in the list of recommended jobs R_k and jobs in the current session history H_s (Zhou et al. 2010). In our experiments, we use the cosine similarity to measure the dissimilarity of two job postings using a raw job vector, which contains 1 if a session interacted with it and 0 otherwise. Again, we use the definition by Vargas and Castells (2011) that takes the recommendation rank as well as the relevance for the current session into account. Hence, we measure the session-based novelty $nov_{session}(R_k|s)$ for the recommendation list R_k of length k for session s by the expected profile distance (EPD) metric:

$$EPD@k = \frac{1}{|S|}\sum_{s\in S}\frac{1}{|R_k||H_s|}\sum_{R_i\in R_k}\sum_{H_j\in H_s} disc(i)p(rel|R_i, s)d(R_i, H_j)$$

Here, H_s is the current history of a session s and $disc(i)$ as well as $p(rel|R_i, s)$ are defined as for the EPC metric for measuring the system-based novelty.

Coverage With coverage (Adomavicius and Kwon 2012; Ludewig and Jannach 2018), we assess how many jobs a recommender approach can cover with its predictions. As such, we additionally report the job coverage of each evaluated algorithm. We define the coverage as the ratio between the jobs that have been recommended and jobs that would be available for recommendation. Here, we make a distinction between coverage types and report the job coverage (1) on the full dataset, i.e., how many of all available jobs can we recommend, and (2) on the test dataset, i.e., how many of the jobs can we recommend that we expect to be interacted with during a session.