Research Proposal

Roan Schellingerhout Francesco Barile Nava Tintarev

December 2022

1 Introduction

A common way for job seekers to find employment, is through their own social network [76]. However, for those with a limited or non-existent network, finding a job can become an insurmountable task. Due to the extremely large amount of possibilities, all in different fields and requiring different skills, finding a job can be compared to finding a needle in a haystack. As a result, the staffing industry has seen a continuous growth over the past decades [15]. Yet, even for experienced recruiters, successfully matching thousands of candidates to thousands of job vacancies can be impossibly challenging. In order to assist in this task, recruitment agencies have increasingly been making use of machine learning to match candidates and job vacancies [37, 38, 56, 55, 82, 85]. The machine learning techniques being used often boil down to recommender systems (RSs), which, in a general sense, aim to recommend an item (e.g., a job) to a user (e.g., a candidate) based on different types of data [30, 60]. While recommender systems are often focused specifically on finding the most suitable item for a user, within recruitment, this tends to be insufficient [62]. Considering the fact that multiple stakeholders are involved in the process of recruitment, recommendations need to be satisfactory for each of them. These stakeholders, which consist of any individual or group that can affect, or is affected by, the recommender system [2], all have distinct goals:

Candidates: individuals looking for a job. Their goal is to get hired for the best possible job.

Companies: businesses that are looking for a new employee. Their goal is to hire the most-fitting candidate for the position.

Recruiters: mediators working at a recruitment agency, whose task it is to match a candidate to a company. Their goal is to balance the needs of companies and candidates.

Developers: people creating the JRS. Their goal is to get the highest possible model performance, while keeping in mind concepts such as fairness and bias.

Since the stakeholders within a multi-stakeholder recommender system (MSRS) often have conflicting goals, it is crucial that they can be presented with some justification of the provided recommendation. Without a proper explanation, it can be difficult to accept the recommendation at face value, and even more difficult to convince another party of its correctness. Unfortunately, RSs, and especially MSRSs, often function as black box models, meaning their decisions are not explainable.

In recent years, however, the field of explainable artificial intelligence (XAI) has seen considerable growth. As a result, techniques have been proposed to make even the most complex deep learning models explainable to a degree. One of such techniques is the use of knowledge graphs (KGs) to enrich the data used by the models [66, 77]. Knowledge graphs, which consist of entities (nodes) connected by relations (edges), allow computers to 'reason' over the data to an extent [11]. This reasoning takes shape as the use of relations between entities, and the knowledge about said relations, to find semantic connections between entities. For example, by evaluating the following statements: (*Joe Biden, hasJob, President*) and (*President, is_a, Person*), the computer can determine that Joe Biden is, in fact, a person. Through the usage of such paths, a recommender system can not only determine new relations (i.e., recommendations), but also explain why they are likely to be relevant.

The aim of this research will be to answer the following research question: *How can knowledge-aware employment recommendations be made explainable for a multi-stakeholder environment?*

1. How can employment data be represented as a knowledge graph?
2. What are the requirements of recommender system explanations for the different stakeholders?
3. What type of knowledge-aware recommender system is suitable for generating explainable recommendations for each stakeholder?
4. To what degree do knowledge-aware explanations meet the needs of different stakeholders?

This research proposal is structured as follows: in section 2 an overview of previous research is provided. After that, in section 3, a methodology is proposed that can be used to answer the aforementioned research questions. Lastly, section 4 consists of a proposed schedule used to execute the methodology.

2 Related work

2.1 Knowledge-aware recommender systems

Traditional recommender systems largely rely on user-item interaction data to make recommendations [63]. While a lot of information can be gathered from interactions, they often tend to be unevenly distributed. While the most popular items in the system can have hundreds of thousands of interactions, the overwhelming majority has often only been consumed a handful of times [47, 58]. Similarly, active users will have interacted with numerous items, while a lot of users will barely have interacted with any. This disparity is especially apparent with brand-new users and items, as those will have no recorded interactions whatsoever: the cold-start problem [13]. One way to alleviate this problem is making use of content-based filtering, in which items are recommended based on their static features [45]. However, this approach can be somewhat limited, as it largely relies on direct matches within the content - for example, when using content-based filtering in the movie domain, only movies with, e.g., the same genre will be considered for recommendation. Additionally, it requires items to be stored in a structured manner, which makes it inflexible and difficult to apply to heterogeneous data.

One way to deal with the aforementioned problems that has received a lot of attention in recent years, is the use of knowledge-aware recommender systems (KARS) [9]. Such recommender systems are based on data that has been enriched using a knowledge graph. A KG consists of a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$ between entities. Formally, we define a KG as such:

$$\{(s,p,o)|s,o \in \mathcal{E}, p \in \mathcal{R}\} \tag{1}$$

Wherein subjects s can be connected to objects o using predicates p to form a directed graph.

Such KGs allow reasoning over data, which can illuminate connections that would not exist otherwise. For example, in the movie domain, some directors have also acted, and vice versa; while in collaborative filtering, this would not lead to a match (as the individual would occur in either the 'actor' field or the 'director' field), in a knowledge graph, both the actor-version and director-version of an individual would have a relation to the same entity (albeit a different relationship type). As a result, movies that would otherwise have no matching attributes, can be connected nonetheless. By making such 'hidden' connections between entities explicit, datasets can be made considerably less sparse [71]. Additionally, KGs store data in an unstructured way, allowing for easy integration of heterogeneous data from multiple sources - which can help reduce sparsity even further. When an aptly exhaustive KG is used, this largely nullifies the cold-start problem [61].

2.1.1 Connection-based models

Connection-based KARS use the discrete connections within the KG in order to make predictions [21]. By introducing specific rules or meta-paths that indicate a good match between a user and an item, recommendations can be generated [57]. For example, a path-based recommendation could be generated as follows: *"if a user liked a movie, they will also like its sequel"*. By then parsing the KG for sequels of the movie the user liked, new recommendations can be generated.

2.1.2 Embedding-based models

In embedding-based KARS, the knowledge graph is no longer discretized, but rather converted to a vector representation using KG-embedding techniques, such as TransE and TransR [8, 21, 42]. For example Huang et al. [28] create key vectors (related to attributes of items) using TransE and value vectors (related to users' opinions on said attributes), that are embedded iteratively. By then matching the value vector of users to the key vectors of items, items that match a user's interest can be found and subsequently recommended.

2.1.3 Propagation-based models

Propagation-based models extend the methodology of embedding-based models by basing an entity's embedding not just on its own properties, but, additionally, that of (its relation to) its (extended) neighbors [21]. A well-known propagation based KARS is the Knowledge Graph Attention Network (KGAT), which uses a KG-enhanced bipartite graph of users and items [77]. KGAT embeds entities based on their own values and that of their 'ego-network', consisting off all entities that it is (indirectly) connected to (up to a certain number of 'hops'). It recursively starts at an entity's most far-away neighbors, and then propagates the embedding values back to the original node. An attention mechanism is used to determine the importance of specific relations to neighbors, and weigh their impact on the final embedding accordingly.

However, by considering an entity's entire ego-network, a lot of noise is introduced in the embeddings - a problem that Sha et al. [66] solve with their Hierarchical Attentive Knowledge Graph Embedding (HAKG). By only focusing on the sub-graph created by performing random walks from the original entity to the interacted item during propagation, they can limit the ego-network to only neighbors that are directly relevant to the recommendation. Each entity in the sub-graph is recursively embedded similarly to KGAT, after which a multi-head attention mechanism is used to convert the individual entity embeddings into a single sub-graph embedding.

2.2 Multi-stakeholder recommender systems

Abdollahpouri et al. [2] define a recommendation stakeholder as *any group or individual that can affect, or is affected by, the delivery of recommendations to users*. While users are generally the most important stakeholder, current literature insufficiently considers providers and the system as the other two main stakeholders.

For multi-stakeholder recommender systems, a recommendation should not just consider the user, but also the multitude of other stakeholders. While most literature limits itself to recommender systems from a single-stakeholder viewpoint, MSRSs are extremely common [1]. For example, some of the most-studied domains for recommendation: e-commerce, music, movies, and restaurants are often only considered from the users' point of view. However, these domains deal with a number of other stakeholders, such as providers and the platform for e-commerce, artists and labels for music, production companies and cinemas for movies, and business owners and delivery people for restaurants. By failing to consider the needs of these additional stakeholders, the real-world utility of the system decreases. More so, due to a lack of consideration, some stakeholders might feel underrepresented within the system, which could lead to growing discontent and possibly even stakeholders pulling out [2].

2.2.1 Implementation and evaluation

A simple training, validation, and test set split based on user interactions (e.g. a user's last interaction goes into the test set, their second-to-last goes into the validation set, and everything else will become the training set) is somewhat unintuitive in a multi-stakeholder setting, as not every stakeholder will then necessarily be included in each set, causing the model to insufficiently fulfill their needs [2]. From a company's perspective, it would make more sense to split the data temporally (the training set consists of all interactions before X date, the validations set of everything between X and Y date, and the test set of everything after Y date), so that every company has interactions in each split.

In order to evaluate a MSRS, in addition to the regular user summary scores (nDCG, accuracy, etc.), metrics related to different stakeholder groups should be considered (this can entail both metrics for different stakeholders altogether, or metrics that consider sub-groups within a stakeholder group, such as different company types) [2]. Optimization can then either be done through combining different objectives into a single optimization problem, or using a pipeline that considers different stakeholders' needs at different points. The most common evaluation metrics for providers are the following [2, 18, 68]:

- Exposure: how often does the provider show up in recommendations;
- Hits: how often does the provider *successfully* show up in recommendations;
- Reach: how many users get at least one recommendation from the provider;
- Target reach: the same as reach, but only for the target audience (e.g., candidates who meet the requirements set by the provider);
- PAccuracy: average accuracy of recommendations for items made by the provider (can be exchange for any other metric m).

While these metrics are not necessarily interpretable by themselves, they allow providers to compare their performance within the recommender system to their competitors. However, not just point estimates are relevant. Especially for providers, getting very little utility from the system could lead to them giving up on it altogether. Therefore, variance and skewness are also important in order to verify that certain providers are not receiving too little utility.

2.2.2 Reciprocal recommender systems

In specific scenarios, such as recruitment, online dating, and social networks, two people are recommended to each other, in which case both parties need to agree on the recommendation for it to come to fruition: reciprocal recommendation. When dealing with reciprocal recommendations, a balance needs to be struck between the wishes of both sides - while a company might perfectly fit a job seeker's desires, the job seeker might not have all the qualifications that the company expects. Additionally, aspects such as trust, privacy, reputation, and personal attraction become more important, as reciprocal recommendations tend to deal with people, not just items. To meet the wishes of both parties in a reciprocal environment, there exist three main approaches:

- The most straight-forward approach, in which the system makes a recommendation for both parties and then aggregates the scores [53]. In general, the harmonic mean has been shown to be the best-performing and most straight-forward aggregation strategy [53];
- An approach adopted from the domain of information retrieval: re-ranking. After having generated the recommendations for one of the parties, each member of the other party can be given a list of suitable options (e.g., members of the other party who cross a threshold score). These options can then be re-ranked based on the preferences of the second party. When re-ranking, an effort should be made to not just improve the recommendation list for the second party, but also to stray as little as possible from the original ranking (e.g., through minimizing KL-divergence between the original and re-ranked list) [44];

- The third approach is the most advanced: optimizing the model for both parties simultaneously using multi-objective optimization. For example, Yıldırım et al. [82] give their model two separate outputs, one for each party, for which they then optimize the model weights simultaneously. Depending on the domain, emphasis can be put on recommendation correctness for one of the parties, depending on the cost of false positives for each side.

2.3 Job recommender systems

2.3.1 Peculiarities in JRSs

As was determined by the European Union in the AI act [12], the usage of AI in recruitment can be considered a high-risk scenario. Due to the large impact that career choices can have on individual's life, as well as the fact that recruitment often deals with large amounts of sensitive data, job recommender systems require a tailored approach compared to less impactful recommender systems [50]. However, current state-of-the-art approaches often fail to make use of such tailored approaches, causing aspects such as fairness, explainability, and reciprocity to be largely ignored in current literature [62]. Although this is obviously cause for concern, the lack of attention does make sense from a pragmatic point of view, due to the large number of limitations that JRSs have to deal with. For example, very few publicly available datasets exist for recruitment, and those that are available, contain a limited amount of usable data (e.g., only containing implicit interaction data, not containing any company names associated with vacancies, etc.) [3, 4, 52, 82]. As a result, optimizing JRSs for performance alone already poses a large challenge - other important aspects have therefore been given limited attention.

However, even when a rich dataset is used, the nature of recruitment data makes job recommendation an especially difficult task. In traditional settings, items can be recommended multiple times (and in some cases even infinitely) to different users. However, in the recruitment domain, each vacancy can only be filled by one (or, sometimes, a few) applicant(s), after which both the vacancy and the accepted candidate(s) are likely to disappear from the system [53]. Not only does this makes common RS approaches (e.g., CF) less feasible, but it also adds additional constraints to the recommendations that can be given by the system. While a specific position may be the best fit for a large number of candidates, the fact that only one of them will ultimately be accepted, means that at most a few dozen candidates should receive it as a recommendation [62]. As a result, considering provider-side fairness by increasing the visibility of unpopular vacancies (while limiting the impact on relevancy) can be beneficial for both the candidates and the companies. More so, when there are no vacancies for which the candidate is likely to be accepted (either due to a lack of suitability or due to an already large number of applicants), it could be optimal to not provide the user with any recommendations at all due to the high cost of unsuccessfully applying for a position [62].

2.3.2 State of the art

Nonetheless, job recommendation has received considerable attention in previous work. The concept of finding an adequate candidate for a job, also known as a person-job fit, has existed for the better part of a century [65]. As time progressed, the methods to do so got steadily more sophisticated, going from systemic analysis of candidates and jobs [78], to statistical tests to determine fitness [10, 17], to, more recently, machine learning and deep learning to find matches [38, 56, 82]. Initially, common recommendation practices, namely collaborative filtering and content-based filtering, were applied to the recruitment domain [46]. However, due to the aforementioned peculiarities of job recommendation, this often led to sub-optimal results. As a result, deep learning approaches have seen a considerable growth in popularity in recent years, largely relying on determining a level of similarity and suitability between the candidate's CV on the one hand, and the job description/posting on the other [7, 56].

2.3.2.1 Factorization Machines

Similarly to regular RS research, factorization machines have been a common technique used to tackle job recommendations. Since interaction data is especially sparse in JRSs, factorization machines' (FMs) ability to incorporate side-features is particularly useful [59]. For one, Leksin and Ostapets [39] used a factorization machine on the RecSys 2016 dataset in order to get near-state-of-the-art performance. They did so by combining latent semantic indexing (LSI), a factorization machine, and a support-vector machine (SVM). More recently, Yıldırım et al. [82] used an FM in combination with a neural network to reach state-of-the-art performance for both company- and candidate-side predictions on their own dataset. They did so by first embedding sparse features into a latent space using an embedding layer, and then having the FM and neural network feed into a company- and candidate-prediction unit separately.

2.3.2.2 Recurrent Neural Networks

However, while FMs are adequate at dealing with structured data, they fall short when it comes to processing unstructured information. Considering the large amount of valuable data contained in CVs and job descriptions, this tends to be detrimental. Following the trends in natural language processing (NLP) research, recently, a significant amount of research was focused on the use of recurrent neural networks (RNNs), especially Gated Recurrent Units (GRUs) and Long Short-Term Memory units (LSTMs), to encode textual features in a latent space. Most popularly, Qin et al. [56] use bi-directional LSTMs to extract word-level representations of CVs and job descriptions, after which they are combined with hierarchical, ability-aware representations to make a prediction (the Ability-aware Person-Job Fit Neural Network: APJFNN). Following their work, multiple improvements have been proposed to increase performance, such as Bian et al. [7] who improved upon the APJFNN by combining both word-level, local representations with text-level, global representations. Later, Qin et al. [55] improved upon their own model by additionally including Latent Dirichlet Allocation (LDA) to discover important topics for individual jobs. More recently, Kumari et al. [35] achieved state-of-the-art performance on their dataset using a Siamese bi-directional GRU, which could determine whether or not a job description and a candidate were similar to each other in a reciprocal way.

2.3.2.3 Convolutional Neural Networks

The main strength of RNNs in job recommendation is the ability to convert textual data into useful numerical features. For this task, different neural architectures, such as convolutional neural networks (CNNs), have also garnered a lot of attention within the domain of NLP [67, 81, 84]. By first converting individual tokens to numerical representations (e.g., using word2vec [51] or GloVe [54]) and then running those representations through different kernels, high-quality text embeddings can be generated [43, 80]. This technique can then be applied to job recommendation in the same way as RNNs. For example, Zhu et al. [85] used two parallel, sequential CNNs to embed résumés and job descriptions individually. The separate embeddings could then be matched to each other in order to determine how similar the candidate was to the job description and vice versa in order to generate recommendations. Le et al. [38] expanded upon this approach by not simply calculating the similarity between the two parties, but also the likelihood of both sides being interested in each other.

2.3.2.4 Transformers

More recently, transformers [75] have been getting increasingly popular for job recommendations due to their high-quality text embedding generation, ability to make use of parallelization during training, and their state-of-the-art performance on many different NLP tasks [26, 70, 83, 86]. Additionally, large pre-trained transformers, such as Google's BERT [16], allow for models to perform commendably, even with limited training data.

These benefits have led many to adopt transformers for job recommendations, including large multinational companies, such as LinkedIn [22]. LinkedIn's job recommender consists of a modular model that can take transformer outputs of multiple text sources (e.g., multiple CVs and multiple job descriptions), based on which it can determine the interaction between the different sources in order to generate target scores. The target scores of different text sources are then reranked to generate a reciprocal recommendation. In their work, they propose LiBERT as the transformer used in the model, which is a few-parameter version of BERT trained on LinkedIn data.

Similarly, Randstad introduced conSultantBERT for candidate-job matching on their own dataset [37]. ConSultantBERT is a multilingual, fine-tuned version of Sentence-BERT designed to match candidates' CVs to job descriptions. While Sentence-BERT is usually fine-tuned by determining the similarity between sentence pairs from the same text, conSultantBERT was trained on sentences from heterogeneous data sources (i.e., CVs and job descriptions in different languages), which still proved to lead to good performance.

However, both Randstad and LinkedIn feed an entire CV and job description into the models at once, trimming it when necessary. As a result, some contextual information can get lost in the process - for example, different sections of a CV could carry different levels of importance. By distinctively encoding each section of a résumé separately, while also giving the model access to the entire CV, Li et al. [40] managed to reach SOTA performance with their BERT-based model. By differentiating between different sections, the model can have a holistic idea of the candidate and job description, while still being context-aware.

Rather than only looking at textual data He et al. [25] created distinct 'fields' of data based the topics contained within them (e.g., separate fields for the candidate's location, skills, education, etc.). Textual features were embedded using a pre-trained version of ALBERT [36], while other feature types were normalized or encoded accordingly. By then finding the interaction between all fields of the candidate and vacancy, a multi-head attention mechanism is used to determine relevance, after which an MLP is used to determine the person-job fitness.

2.3.3 Shortcomings

While the aforementioned state-of-the-art models manage to achieve great performance on a plethora of different datasets, each of them falls short in one aspect or another. For one, only a few models allow for any sort of explainability w.r.t. the recommendations that they make. Whenever explainability is available, it is often limited either because the generated explanations convey very little actual information (such as in [38, 82]), or because the authors fail to consider the needs of the different stakeholders (such as in [74]). While a feature attribution map can be very useful for a developer, a job seeker or a recruiter is likely to struggle making sense of it [69].

Additionally, data sparsity continues to be a major detriment in many scenarios. Due to the fact that candidates and vacancies leave the system upon being successfully recommended to each other, very little interaction data exists in most of the datasets used. Little has been done in previous work to counteract this sparsity (e.g., using additional data sources for enrichment, either in the form of combining structured datasets or making use of knowledge bases). Although some knowledge-based job recommender systems have been proposed for job recommendation, they have often been limited in their approach, sticking mostly to path-based approaches [23, 49, 74].

2.4 Explainability in RSs

While achieving the highest possible accuracy is obviously desirable for a lot of recommender systems, their user-focused applications make it so that other factors, such as explainability, should also be considered - more so than in other machine learning domains. However, explainability is not a one-dimensional concept. Depending on the context wherein the RS is being used, explainability can have multiple different goals, including transparency, scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction [72]. These goals are often (partially) incompatible, meaning that a balance needs to be struck based on which goals are most important in the given domain. Depending on the items and users interacting with the recommender system,

Model type	Model Name	Authors	Dataset	Year
FM-based	N/A	Leksin and Ostapets	RecSys 2016	2016
FM-based	biDeepFM	Yıldırım et al.	kariyer.net	2021
RNN-based	APJFNN	Qin et al.	BOSS Zhipin	2018
RNN-based	TDGMN	Bian et al.	BOSS Zhipin	2019
RNN-based	TAPJFNN	Qin et al.	Baidu Inc.	2020
CNN-based	PJFNN	Zhu et al.	Private	2018
CNN-based	IPJF-JT	Le et al.	BOSS Zhipin	2019
Transformer-based	CIJA	Li et al.	CRC	2020
Transformer-based	LiBERT	Guo et al.	LinkedIn	2020
Transformer-based	conSultantBERT	Lavi et al.	Randstad	2021
Transformer-based	MUFFIN	He et al.	Zhoapin	2021

Table 1: An overview of the current state of the art for job recommender systems.

it might be more desirable to give highly transparent explanations, possibly at the cost of efficiency, or to opt for more general explanations to improve persuasiveness without improving transparency [73].

Furthermore, a conscious decision needs to be made regarding the way in which the explanations are conveyed to the user. For example, explanations can be shown textually, visually, or in some type of hybrid form. While textual explanations have been shown to be the most persuasive and effective [34], they also tend to lead to a lesser degree of satisfaction [69]. Even when only considering textual or visual explanations, the way in which they are conveyed to the user can have a significant impact on the different goals [27]. As a result, the way in which recommendations are explained needs to be carefully considered during the design phase of the RS, and needs to be properly evaluated to make sure it is positively affecting the goals of the system.

2.4.1 Evaluating explanations

For each type of goal, Tintarev and Masthoff [72] define specific ways to evaluate the provided explanations:

Transparency: in order to evaluate transparency, users could be asked if they understand how the explanations came to be, or tasked to correctly influence the recommendation through changing specific variables (such as in Szymanski et al. [69]);

Scrutability: when evaluation scrutability, users can be given a task where they are supposed to scrutinize the model (e.g. "stop receiving X recommendations"). Then, objective measures (task correctness and required time), as well as questionnaires, can be used to determine scrutability;

Trust: when evaluating user trust, questionnaires can be very helpful. However, questionnaire answers do not always reflect real-world behavior. Online testing could be used to detect byproducts, such as loyalty and increased sales, which indirectly relate to trust;

Persuasiveness: persuasiveness can be measured by looking at the difference in likelihood of selecting an item in an environment with explanations and one without them. Additionally, users could be tasked to re-rate items after having received an explanation to see if their perception of the items increased (i.e., they got persuaded of its relevance);

effectiveness: evaluating effectiveness can be extra difficult, because it relies on whether or not the recommended item was indeed relevant, which can only be judged after consumption. For example, users could be tasked to give a rating before and after consuming the item - if their rating remained the same, it means the explanations was effective, since it gave the user a good idea of what to expect;

efficiency: measuring efficiency is rather straight-forward, as one could simply measure how much time/how many interactions were required before a suitable item was found;

satisfaction: since satisfaction is a very subjective concept, the most straight-forward way of evaluating it, is to ask users if they feel satisfied with the process and the recommended product.

2.5 Explainability in KGBRSs

2.5.1 Path-based models

Generating explanations for path-based models is rather straight-forward, as they are inherently based on (meta-)paths within the knowledge graph. By simply showing the user the paths that were considered when making the recommendation, an explanation can be generated [57].

2.5.2 Embedding-based models

When creating user embeddings, use can be made of attention mechanisms [5] in order to determine what attributes are the most important for the embedding. As Wiegreffe and Pinter [79] concluded, attention values can largely be used as explanations. More specifically, attention values provide a *plausible* explanation - this is not necessarily a *faithful* (i.e., the 'correct') explanation, but can generally be considered to be sufficient.

2.5.3 Propagation-based models

During embedding propagation, attention mechanisms can be used in order to determine the importance of each relation within the ego-network or sub-graph on the final embedding of the entity [66, 77]. To then explain a recommendation, the path (or paths) with the highest attention values can be 'walked', as those had the largest impact on the final embedding and recommendation.

2.6 Explainability in RRSs

Explaining reciprocal recommendations tends to be more difficult than regular recommendations, as the preferences of both parties need to be considered. Kleinerman et al. [32] determined that explanations that consider both parties outperform one-sided explanations in high-cost scenarios (such as recruitment). Especially explanations based on specific feature values are useful, although only a limited number of features should be included to prevent information overload. In high-cost scenarios, explanations should not stay limited to personal preferences (e.g. 'you should apply for this job because you want a company that has X attributes'), but should also incorporate an explanation on why the other party is likely to agree (e.g. 'they are likely to accept you, because they are looking for a candidate with Y skills'). However, due to privacy concerns, it can be difficult to reveal the recommended candidate/company's specific preferences to the receiver of the recommendation [33].

Although it offers a less comprehensive explanation, Kleinerman et al. [32] found that not mentioning any specific preferences, but simply stating that the other party is likely to be interested in you as well, was a good solution to this issue. Alternatively, counterfactual explanations could be used when mentioning the other party's preferences. Counterfactual explanations never explicitly mention the other party's preferences, but merely refer to each party's own choices, and how different choices would have changed the outcome [14].

2.6.1 Explainability in JRSs

In job recommender systems specifically, (reciprocal) explainability has largely gone unexplored. While some previous work has incorporated some degree of explainability within their JRSs, the explanations are often limited, and seem to have been included as an afterthought [38, 74, 82]. Even when explainability has been included, authors often fail to consider both parties, tailoring the explanations to the job seekers only. In a high-risk, high-impact domain such as recruitment, one could argue that reciprocal, easy-to-understand explainability should be at the core of the models' design. Where previous research mostly falls short, is in the understandability of their explanations: while their models can technically explain some part of their predictions, the explanations tend to be unintuitive and/or limited, either staying too vague ([38, 74]) or being hard to understand ([82]). When dealing with users with limited AI knowledge, such as recruiters, job seekers, and most companies, having clear, straight-forward explanations is crucial [64, 69]. To accomplish this, structured requirements engineering will have to be conducted in order to determine the preferences of all stakeholders, after which explainable JRSs will need to be designed with those requirements as a starting point.

When considering the goals set out by Tintarev and Masthoff [72], certain goals are more important than others in the domain of recruitment. For one, trust is usually the most crucial aspect of a JRS, as career decisions can have a considerable impact on an individual's life. As a result, candidates are unlikely to accept recommendations when they do not fully trust the system [29]. Secondly, effectiveness is highly important, as the process of applying and being accepted for a position require a great deal of effort from all parties involved. If that effort ends up going to waste due to unreliable explanations and recommendations, the usability of the system deteriorates strongly [62]. Furthermore, persuasiveness is critical in a reciprocal environment, as not one, but two sides need to be convinced that the recommendation is a correct one [32]. Lastly, scrutability tends to be important in order to account for biases. Recommendations that were made based on erroneous, biased assumptions (e.g., stereotypes that protrude from the data) should be easy to spot and account for [20]. Not only will this improve trustworthiness, but it will also help in improving fairness by bringing biases to light.

While satisfaction obviously needs to be taken into account when designing any system, it plays a less significant role in JRSs than in most other domains (such as movie, music, and video recommendation). Considering the system itself will primarily be used by recruiters, it needs to be functional and easy to use, rather than entertaining. Efficiency also tends to be less relevant in JRSs, as high-impact decisions such as an individual being hired often require a substantial amount of consideration regardless - i.e., it is generally acceptable if a suitable recommendation takes some time to be provided, as long as it ends up being correct [62]. Similarly, transparency is not as important in JRSs, as the main stakeholders involved in the process, candidates, companies, and recruiters, tend to have a limited amount of knowledge on recommender systems. Therefore, providing them with a full explanation of the inner-workings of the model could do more harm (by causing confusion and information overload) than good [64].

2.6.2 Evaluating explainability in JRSs

Due to the asymmetric nature of job recommendations, evaluating explanations comes with additional complexities. While explanations might be well-suited for one of the parties, they might be insufficient or overly complicated for the others [64, 69]. As a result, the explanations need to be evaluated separately for each stakeholder. The evaluation approach often needs to be customized for each party involved, as their way of interacting with the system can differ.

While the exact implementation often differs between stakeholders, the general methodology can often remain the same [48] as with general RRSs. For example, in order to evaluate efficiency and trust, offline (simulated) or online user testing, in combination with questionnaires and interviews can be used. By letting a sufficient sample of candidates, recruiters, and company representatives try out the (simulated) system, both with and without explanations, the more 'objective' goals, such as persuasiveness, effectiveness, and efficiency, could be measured. By requiring the different users to perform additional tasks during their try-out, transparency and scrutability can additionally be evaluated. Lastly, the most subjective goals, namely trust and satisfaction, could be measured using questionnaires and interviews.

While this might seem straight-forward, two additional snags need to be considered: (I.) different stakeholders will prefer different types of explanations, and (II.) the try-outs, tasks, and questionnaires/interviews would need to differ for each type of stakeholder.

To tackle the first issue, preliminary semi-structured, task-based interviews could be conducted in order to determine the preferences that each party holds [48, 69]. For example, candidates and companies are likely to have more specific domain knowledge than recruiters, while recruiters could be better at determining what 'types' of candidates fit a company's goals and values. Therefore, recruiters may prefer more general rules of thumb as explanations, and candidates/companies might prefer very specific explanations based on specific skills and prior experience.

The second issue, in part, relies on the solution used for the first issue. Depending on the type of explanations provided, the tasks and interviews should differ. However, apart from those intricate differences, the structure of the try-outs, tasks, and questionnaires/interviews could largely stay the same across stakeholders. For example, the specific tasks should be adjusted to fit the goals of the stakeholder - a recruiters could be tasked to find the top 5 most promising candidates for a vacancy, while a candidate could be tasked to find some vacancies that match their interests and are likely to accept their application. While these tasks are obviously distinct, the measures used to determine persuasiveness, efficiency, etc. can stay largely the same (e.g., in both cases the relative difference in time required to complete the task with and without explanations can be measured).

3 Proposed methodology

In order to answer the main research question, as well as the aforementioned sub-questions, a number of steps need to be undertaken. This section outlines the required preliminary steps, as well as the proposed approaches to answer each question.

3.1 RQ1 - Knowledge graph creation

3.1.1 Dataset

While a handful of career datasets are publicly available, they tend to be limited and of poor quality [62]. Although they are still useful to evaluate model generalizability, using them as the sole training data would be undesirable. As a result, the first step of our research will be the creation of a career dataset based on Randstad NV's (Randstad) data. As the largest employment agency in the world, Randstad has an exhaustive dataset consisting of numerous types of data. Randstad stores general information on job seekers, such as their employment history, education history, skills, spoken languages, CVs, etc. Similarly, for job vacancies, it stores attributes such as the company offering the position, the salary indication, the location of the job, and so forth. Additionally, Randstad has extremely high-quality interaction data, which is generated based on professional recruiters' linkage of candidates to job vacancies.

Considering the large amount of highly private data contained within some attributes, this dataset is, as of now, completely private to Randstad. However, making the dataset publicly available, e.g. in the form of a challenge, could be beneficial to Randstad. By incentivizing researchers to train and test new models on the data, a clear state of the art could be determined. Randstad could then make use of the state-of-the-art models in their everyday production, leading to better performance for their customers. In order to publish the dataset, it would need to be in accordance with data privacy laws, such as the GDPR [19]. For this, the data would need to be anonymized, e.g. through the use of k-anonymization.

3.1.2 Knowledge graph

While the quality of the created dataset will be high, it will still be extremely sparse. The overwhelming majority of candidates only have a single job in their employment history, and vacancies have often only been linked to a single candidate. To alleviate this sparsity, Randstad's data can be converted to a knowledge graph by linking it to existing ontologies that are publicly available online (e.g. Wikidata[1], DBpedia[2], and O*NET OnLine[3]). After linking the interaction data to existing ontologies, more sophisticated links between different interactions can be determined, leading to a decrease in data sparsity.

For example, Randstad's data includes the name of the company that created a job vacancy. While this makes it possible to find relations between jobs offered by the same employer, similarities between separate employers remain hidden. By linking company names to an existing ontology, more implicit relations can become explicit - e.g., while Target and Walmart are completely separate entities, they are both department stores that predominantly operate in the United States. The fact that such higher-level commonalities are retrievable in knowledge graphs is highly beneficial for JRSs.

3.2 RQ2 - Stakeholder explainability requirements

To ensure that the explanations generated by the recommender system end up meeting the requirements of the different stakeholders, the explanations themselves will be used as a starting point, based on which the recommender system itself will be designed. In order to discover the preferences of different stakeholders, semi-structured interviews will be conducted using example explanations. During these semi-structured interviews, the participants will be tasked to answer substantive questions based on the provided explanations, as well as to indicate what aspects of different explanations they prefer. This way, it will be possible to determine which explanation types are most satisfactory and effective for each stakeholder. Each of the three main stakeholders will be represented by $\mathbf{N}(= 10?)$ individuals in order to have a sufficient sample. Each participant will be interviewed about each explanation type. The explanation types that will be examined in this study will be the following:

[1]https://www.wikidata.org/
[2]https://www.dbpedia.org/
[3]https://www.onetonline.org/

Textual: a short text that explains which features contributed to the recommendation in what way;

Feature attribution: a visualization (usually a bar chart) that shows which features were most important to the model when creating the explanation (fig. 1);

Paths: a visualization of paths in a knowledge graph. In our case, this consists of the paths within the candidate-vacancy sub-graph (fig. 2);

Hybrid: a combination of the three above types. Concretely, a hybrid explanation consists of one or more visualizations supported by some explanatory text.

The participants will be interviewed about all four explanation types once. The order in which the explanation types are shown will be randomized to minimize potential bias (as later explanations could develop an advantage due to participants getting more experienced). As a result, each explanation type will have a total of 30 evaluations, split across the three stakeholder types.

For all three stakeholder types, we expect to see similar trends in explanation preferences. Out of the non-hybrid options, textual explanations are expected to be the easiest to interpret by the participants, leading to the highest percentage of correctly answered questions. However, the feature attribution maps are predicted to be the most-preferred one, as they require less effort to analyze compared to texts and paths. Although path-based explanations generally provide the most 'faithful' explanations, we hypothesize that participants will find them difficult to work with. Therefore, the hybrid explanation is expected to be the best option, as it can combine the benefits of all three alternatives, while limiting their drawbacks.

3.3 RQ3 - Model architecture

In order to make the reciprocal, explainable explanations, we propose the Occupational Knowledge graph-based Recommender using Attention (OKRA)

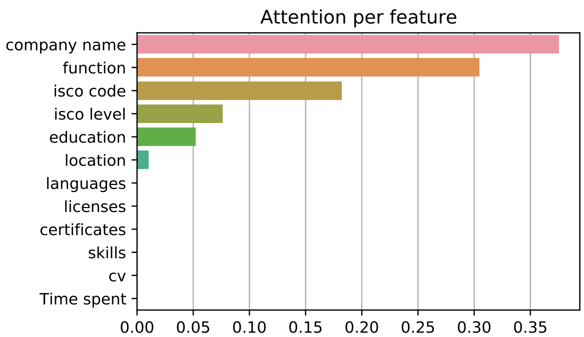

Figure 1: An example of a feature attribution map used as an explanation for a recommendation.

(fig. 3), which is an explainable [77, 79], reciprocal [82] propagation-based knowledge graph recommender system [66]. The input for this model consists of sub-graphs created from the KG-enriched candidate/vacancy interaction graph. The sub-graphs are created using the k-random walks algorithm [41, 66], which aims to find paths between the candidate and target vacancy of less than k steps (or 'hops'). Each node within the sub-graph, $\mathcal{G}_{c,v}$, then gets embedded using a propagation-based embedding method. Candidate and vacancy embeddings are initialized using the conSultantBERT embeddings [37] of their CV or job description respectively. Afterwards, each node is updated propagatively based on its relations to its neighbors. The impact of a node's neighbors' values on its own embedding value is determined by an

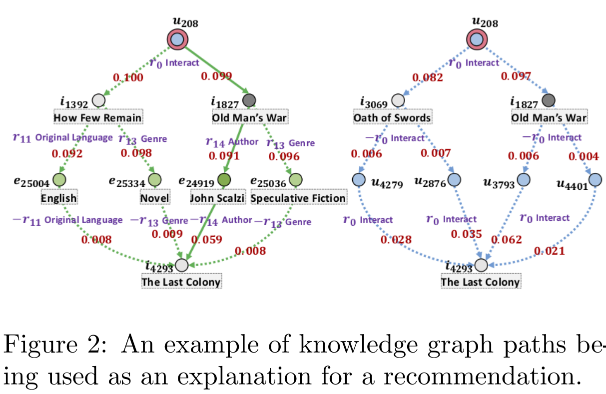

Figure 2: An example of knowledge graph paths being used as an explanation for a recommendation.

attention module. The relational embedding value $e \in \mathbb{R}^d$ of a given node e_g to its neighbor e_n, with dimension d, at propagation layer l is calculated as follows:

$$r_{g\leftarrow n}^{(l)} = W_1^{(l)} \cdot \left(e_n^{(l-1)} \oplus r_{g,n}\right) \tag{2}$$

Here, $W_1^{(l)}$ is the trainable weight matrix at propagation layer l, $e_n^{(l-1)}$ is the embedding value of the current node's neighbor at propagation layer $l-1$, $r_{g,n}$ is the embedding of the relation between the two nodes, and $\oplus$ indicates the concatenation operation.

The embedding values for each node e_n at propagation layer l are then calculated using the following equation:

$$e_g^{(l)} = ReLU\left(W_2^{(l)} e_g^{(l-1)} + \sum_{\forall e_n \in \mathcal{N}_g} \alpha(M, e_n, g, n)\right) \tag{3}$$

in which $W_2^{(l)}$ is a second trainable weight matrix, $e_g^{(l-1)}$ is the node's current embedding value, $\mathcal{N}_g$ is the node's neighborhood, $M \in \mathbb{R}^{d\times|\mathcal{G}_{(c,v)}|}$ whose rows are given by the embedding vector of each node in $\mathcal{G}_{(c,v)}$, and

$$\alpha(M, e_n, g, n) = softmax\left(w_1 \cdot \sigma\left(W_3 M^T\right)\right) \cdot r_{g\leftarrow n}^{(l)} \tag{4}$$

where w_1 and W_3 are a trainable weight vector and matrix respectively.

As a result, each node's embedding values are dependent on its own embedding, as well as the weighed embeddings of its (relation to its) neighbors. Because each node is therefore influenced by its neighbors, all nodes are indirectly impacted by the entire sub-graph. However, the attention mechanism allows the model to determine which relations are important to the current node and which are not.

After having determined the embedding values of each individual node in the sub-graph, the embeddings are concatenated, after which the model uses two separate multi-head attention modules to embed the sub-graph as a whole. These multi-head attention modules combine the individual node embeddings in

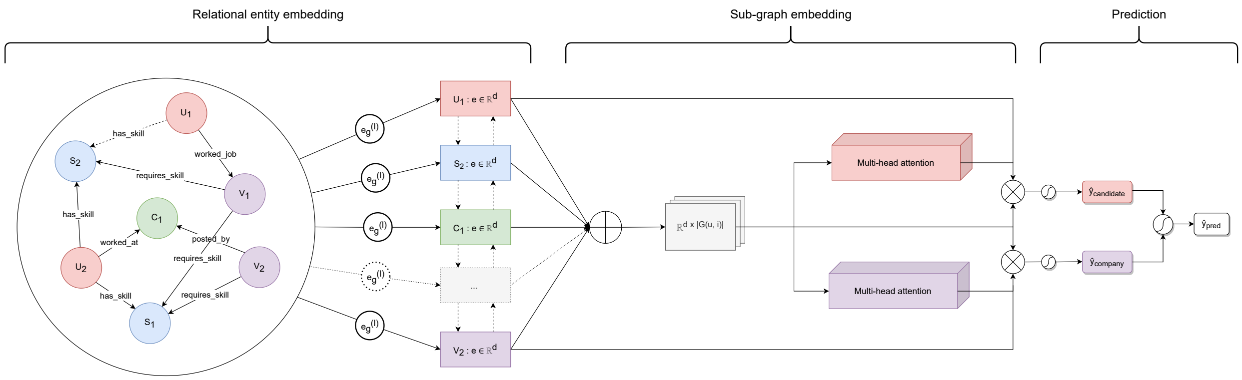

Figure 3: An overview of the proposed model architecture, OKRA. **W.I.P.**

two separate ways - once for the candidate and once for the company. The rationale behind this, is that different factors hold different levels of importance depending on whether someone is looking for a job or for an employee. Thus, the multi-head attention modules allow the model to attend to specific nodes differently, thereby attributing different levels of importance to them. The model then outputs two separate ratings: one predicted 'match' score for the candidate and one for the company. These two scores are then balanced using their harmonic mean, which generates a final predicted matching score, based on which candidates and companies can be ranked.

3.3.1 Explainability

Because of the use of attention modules throughout the entire model, it is inherently explainable. Initially, the relational embedding's use of attention when updating node embedding values makes it so that, for each node, the impact of all its neighbors and relations on its own embedding are retrievable. These attention weights can therefore be used to find the most important paths between individual nodes. When combining this with the multi-head attention weights, which specifically focus on which *nodes* were important for the prediction, 'explanatory paths' can be generated. Using the relational attention weights, the 'most important' paths from the candidate to the vacancy can be retrieved, which can be reweighed using the multi-head attention weights to give more important nodes more impact on the explanation. This can be done separately for the candidate and the company, leading to unique explanations for the two main stakeholders.

3.3.2 Optimization

The model will be trained on 80% of Randstad's dataset ($N =$ ...), which will be sampled randomly (while assuring a proper distribution of candidates and companies). During training, model parameters will be optimized using the Adam optimizer [31] based on the with cross-entropy loss between the final prediction score compared to the ground truth interaction value.

3.3.3 Model evaluation

The remaining 20% of the data will be split in half, after which 10% of the data will be used as a validation set ($N =$ ...), and another 10% as a test set ($N =$ ...). Hyperparameter tuning will be performed on the evaluation set using randomized grid search [6].

To evaluate general model performance, general classification metrics - accuracy, recall, precision, and the F1-score - will be used. These metrics give an indication of how well the model can detect matches between candidates and companies, but they are generally naive within the domain of job recommendation. For candidates, it is not just important if a recommendation is a good match; factors such as likelihood to be hired and fairness should also be considered. Since the process of applying for a job requires a significant amount of effort, a candidate cannot be expected to apply for every recommended vacancy. Thus, the model should additionally make sure to properly spread vacancy recommendations across candidates, and vice versa. As a result, candidates will have a higher probability of actually getting hired for the positions they apply for, and companies are more likely to get their vacancies filled.

For the companies, exposure, hits, reach, target reach, and PAccuracy will also be calculated to ensure proper provider-side performance. For these metrics, the distribution of values across companies will be especially considered to ensure company-side fairness is respected. Candidate-side fairness will be evaluated using standard fairness metrics, namely demographic parity and equalized odds [24].

The OKRA's scores will then be compared to a multitude of baselines on the aforementioned metrics; three simple baselines: ... and four deep learning alternatives: conSultantBERT, biDeepFM, KGAT, and HAKG. Additionally, to evaluate model generalizability, an issue often overlooked in job recommendation [62], models will be tested on three separate datasets: Randstad's dataset, the CareerBuilder 2012 dataset, and the Zhoapin dataset.

3.4 RQ4 - Evaluation of explanations

In order to evaluate the efficacy of the explanations generated by OKRA, exhaustive user testing will be conducted for the candidates, companies, and recruiters. This user testing will take shape in the form of a between-subject study, wherein stakeholders interact with a simulated environment (see fig. 4) in which they will need to execute a clear task: select the most suitable recommendation (i.e., the one that fits the candidate/companies preferences the most, and is most likely to lead to a hire) from the list of generated recommendations.

For the three main stakeholders, the environment will look mostly the same, but contain different types of information.

Firstly, for recruiters, both the candidates and vacancies need to be represented, as well as an explanation on why they are a good match. Thus, the environment will have two modes: one for matching a candidate to a vacancy (candidate-view), and one for matching a vacancy to a candidate (company-view). As a result, recruiters will be able to use the system regardless of the party requiring their assistance. Depending on the inquirer, the environment will firstly show an overview of either the candidate or the job vacancy, so the recruiter will be able to make an informed decision. Secondly, a list of recommendation will be shown, either consisting of suitable candidates or vacancies - accompanied by their popularity/availability. When the recruiter selects one of these recommendations, the exact information of that recommendation will be shown (so either information about the recommended candidate or vacancy) and an explanation on why that candidate/vacancy is a good fit.

For the candidates and companies, the environment will be similar, but simpler and limited to a one-way perspective. In other words, candidates will be able to browse recommended vacancies and inspect the accompanying explanations, but will not be able to switch to a company-view and vice versa. Candidates will receive more limited information on the vacancies, as not all information Randstad has about companies is publicly available. Similarly, companies will not be able to view all information Randstad has on candidates, considering some information is private. The explanations shown in the candidate- and company-view will be the respective explanations generated by the multi-head attention modules in OKRA, accompanied by a summarized version of the other party's explanation to allow for reciprocity without giving away too much information.

Per type stakeholder, the participants will be split into two distinct groups. The individuals in the different groups will be faced with an altered version of the simulated environment. The first group will be able to interact with the full environment as specified. The second group will only be able to observe the list of recommendations and the candidate/vacancy data, with no explanations whatsoever. This is how Randstad's current system operates, and is therefore

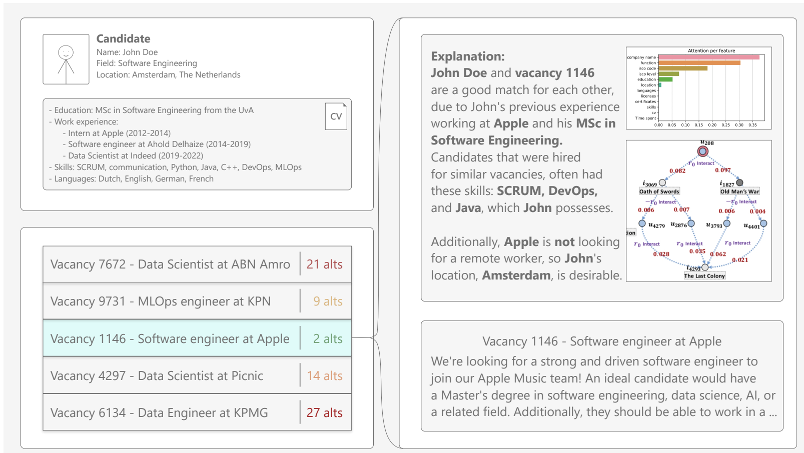

Figure 4: A prototype of what the simulated environment will look like (recruiter-view).

the baseline that OKRA's explanations will attempt to beat. In everyday operation, the list of recommendations generated by Randstad's current system contributes to 28% of matches.

Each participant will be asked to think out loud while performing the tasks in order to get a better understanding of their interpretation of the different explanation types, as well as to detect possible ambiguities and errors. The objective goals of scrutability, persuasiveness, effectiveness, and efficiency, will be measured using the regular methods: task correctness, likelihood to accept a recommendation, [way to measure effectiveness in this scenario $-->$ perhaps predicting how long the candidate would stay?], and time to find a recommendation.

The subjective goals - transparency, trust, and satisfaction - on the other hand, will be measured using a combination of questionnaires and interview questions.

4 Schedule

1. Creating the dataset
2. Interviewing recruiters, companies, and candidates to determine what type of explanations are suitable
3. implementing SOTA models
4. making my own explainable, reciprocal model according to the requirements found in 2.
5. evaluating the model from a multi-stakeholder perspective - when does it perform well?
6. evaluating the explanations in a:
   - experimental scenario (similar to [32])
   - real-world scenario (similar to [69])

5 Talking points

- VR teambuilding thing for XAI

Possible data sources:

- xing.com, from the RecSys challenge 2017 (could ask for permission)
- kariyer.net
- 2012 CareerBuilder competition $-->$ lacks company names, but could be extracted from descriptions? Also has very little data on users (just location, education, and previous job titles)

- Zhilian $-->$ has interaction data, work experience/skills, education, etc. However, everything is in Chinese and not linked to ISCO. Also mostly unstructured data.

References

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. "Recommender systems as multistakeholder environments". In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization.* 2017, pp. 347–348.

[2] Himan Abdollahpouri et al. "Multistakeholder recommendation: Survey and research directions". In: *User Modeling and User-Adapted Interaction* 30.1 (2020), pp. 127–158.

[3] Fabian Abel et al. "Recsys challenge 2016: Job recommendations". In: *Proceedings of the 10th ACM conference on recommender systems.* 2016, pp. 425–426.

[4] Fabian Abel et al. "Recsys challenge 2017: Offline and online evaluation". In: *Proceedings of the eleventh acm conference on recommender systems.* 2017, pp. 372–373.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[6] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[7] Shuqing Bian et al. "Domain adaptation for person-job fit with transferable deep global match network". In: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP).* 2019, pp. 4810–4820.

[8] Antoine Bordes et al. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems* 26 (2013).

[9] Robin Burke. "Knowledge-Based Recommender Systems". In: *ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS.* Marcel Dekker, 2000, p. 2000.

[10] David F Caldwell and Charles A O'Reilly III. "Measuring person-job fit with a profile-comparison process." In: *Journal of applied psychology* 75.6 (1990), p. 648.

[11] Xiaojun Chen, Shengbin Jia, and Yang Xiang. "A review: Knowledge reasoning over knowledge graph". In: *Expert Systems with Applications* 141 (2020), p. 112948.

[12] European Commission. *Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts.* Apr. 2021. URL: `https://eur-lex.europa.eu/legalcontent/EN/TXT/?qid=1623335154975%5C&uri=CELEX%5C%3A52021PC0206`.

[13] Michelle Keim Condliff et al. "Bayesian mixed-effects models for recommender systems". In: *ACM SIGIR.* Vol. 99. Citeseer. 1999, pp. 23–30.

[14] Giandomenico Cornacchia et al. "Explanation in multi-stakeholder recommendation for enterprise decision support systems". In: *International Conference on Advanced Information Systems Engineering.* Springer. 2021, pp. 39–47.

[15] Statista Research Department. *Staffing industry: Global revenue.* Oct. 2022. URL: `https://www.statista.com/statistics/624116/staffing-industry-revenue-worldwide/`.

[16] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[17] Karen Holcombe Ehrhart and Guido Makransky. "Testing vocational interests and personality as predictors of person-vocation and person-job fit". In: *Journal of Career Assessment* 15.2 (2007), pp. 206–226.

[18] Michael D Ekstrand et al. "Exploring author gender in book rating and recommendation". In: *Proceedings of the 12th ACM conference on recommender systems*. 2018, pp. 242–250.

[19] European Commission. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[20] Mouadh Guesmi et al. "Open, Scrutable and Explainable Interest Models for Transparent Recommendation." In: *IUI Workshops*. 2021.

[21] Qingyu Guo et al. "A survey on knowledge graph-based recommender systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[22] Weiwei Guo et al. "DeText: A Deep Text Ranking Framework with BERT". In: *International Conference on Information and Knowledge Management, Proceedings* (Oct. 2020), pp. 2509–2516. DOI: 10.1145/3340531.3412699.

[23] Anika Gupta and Deepak Garg. "Applying data mining techniques in job recommender system for considering candidate job preferences". In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. 2014, pp. 1458–1465.

[24] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems* 29 (2016).

[25] Miao He et al. "Self-Attentional Multi-Field Features Representation and Interaction Learning for Person-Job Fit". In: *IEEE Transactions on Computational Social Systems* (2021).

[26] Pengcheng He et al. "Deberta: Decoding-enhanced bert with disentangled attention". In: *arXiv preprint arXiv:2006.03654* (2020).

[27] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. "Explaining collaborative filtering recommendations". In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 2000, pp. 241–250.

[28] Jin Huang et al. "Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks". In: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: Association for Computing Machinery, 2018, pp. 505–514. ISBN: 9781450356572. DOI: 10.1145/3209978.3210017. URL: https://doi.org/10.1145/3209978.3210017.

[29] Dean Karlan et al. "Trust and social collateral". In: *The Quarterly Journal of Economics* 124.3 (2009), pp. 1307–1361.

[30] Jussi Karlgren. *An algebra for recommendations: Using reader data as a basis for measuring document proximity*. 1990.

[31] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[32] Akiva Kleinerman, Ariel Rosenfeld, and Sarit Kraus. "Providing explanations for recommendations in reciprocal environments". In: *Proceedings of the 12th ACM conference on recommender systems*. 2018, pp. 22–30.

[33] Irena Koprinska and Kalina Yacef. "People-to-people reciprocal recommenders". In: *Recommender Systems Handbook*. Springer, 2022, pp. 421–446.

[34] Pigi Kouki et al. "Personalized explanations for hybrid recommender systems". In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 2019, pp. 379–390.

[35] Tulika Kumari, Ravish Sharma, and Punam Bedi. "Siamese Bi-Directional Gated Recurrent Units Network for Generating Reciprocal Recommendations in Online Job Recommendation". In: *International Conference on Innovative Computing and Communications*. Springer. 2023, pp. 257–269.

[36] Zhenzhong Lan et al. "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (2019).

[37] Dor Lavi, Volodymyr Medentsiy, and David Graus. *conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers*. 2021. URL: https://tika.apache.org/.

[38] Ran Le et al. "Towards effective and interpretable person-job fitting". In: *International Conference on Information and Knowledge Management, Proceedings* (Nov. 2019), pp. 1883–1892. DOI: 10.1145/3357384.3357949. URL: https://doi.org/10.1145/3357384.3357949.

[39] Vasily Leksin and Andrey Ostapets. "Job Recommendation Based on Factorization Machine and Topic Modelling". In: *Proceedings of the Recommender Systems Challenge on - RecSys Challenge '16* (2016). DOI: 10.1145/2987538. URL: http://dx.doi.org/10.1145/2987538.2987542.

[40] Changmao Li et al. "Competence-level prediction and resume & job description matching using context-aware transformer models". In: *arXiv preprint arXiv:2011.02998* (2020).

[41] Rong-Hua Li et al. "On random walk based graph sampling". In: *2015 IEEE 31st international conference on data engineering*. IEEE. 2015, pp. 927–938.

[42] Yankai Lin et al. "Learning entity and relation embeddings for knowledge graph completion". In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

[43] Genggeng Liu et al. "An intelligent CNN-VAE text representation technology based on text semantics for comprehensive big data". In: *arXiv preprint arXiv:2008.12522* (2020).

[44] Weiwen Liu and Robin Burke. "Personalizing fairness-aware re-ranking". In: *arXiv preprint arXiv:1809.02921* (2018).

[45] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends". In: *Recommender systems handbook* (2011), pp. 73–105.

[46] Yao Lu, Sandy El Helou, and Denis Gillet. "A recommender system for job seeking and recruiting website". In: *Proceedings of the 22nd International Conference on World Wide Web*. 2013, pp. 963–966.

[47] Linyuan Lü et al. "Recommender systems". In: *Physics reports* 519.1 (2012), pp. 1–49.

[48] Ana Lucic et al. "A multistakeholder approach towards evaluating ai transparency mechanisms". In: *arXiv preprint arXiv:2103.14976* (2021).

[49] Jorge Martinez-Gil, Alejandra Lorena Paoletti, and Mario Pichler. "A novel approach for learning how to automatically match job offers and candidate profiles". In: *Information Systems Frontiers* 22.6 (2020), pp. 1265–1274.

[50] Yoosof Mashayekhi et al. "A challenge-based survey of e-recruitment recommendation systems". In: *arXiv preprint arXiv:2209.05112* (2022).

[51] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[52] Ravita Mishra and Sheetal Rathi. "Enhanced DSSM (deep semantic structure modelling) technique for job recommendation". In: *Journal of King Saud University-Computer and Information Sciences* (2021).

[53] Iván Palomares et al. "Reciprocal recommender systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation". In: *Information Fusion* 69 (2021), pp. 103–127.

[54] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[55] Chuan Qin et al. "An enhanced neural network approach to person-job fit in talent recruitment". In: *ACM Transactions on Information Systems (TOIS)* 38.2 (2020), pp. 1–33.

[56] Chuan Qin et al. "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach". In: *The 41st international ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 25–34.

[57] Antonio Rago et al. "Argumentative explanations for interactive recommendations". In: *Artificial Intelligence* 296 (2021), p. 103506.

[58] Al Mamunur Rashid, George Karypis, and John Riedl. "Influence in ratings-based recommender systems: An algorithm-independent approach". In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM. 2005, pp. 556–560.

[59] Steffen Rendle. "Factorization machines". In: *2010 IEEE International conference on data mining*. IEEE. 2010, pp. 995–1000.

[60] Elaine Rich. "User modeling via stereotypes". In: *Cognitive science* 3.4 (1979), pp. 329–354.

[61] Rosa M Rodríguez et al. "Using linguistic incomplete preference relations to cold start recommendations". In: *Internet Research* (2010).

[62] Corné de Ruijt and Sandjai Bhulai. "Job recommender systems: A review". In: *arXiv preprint arXiv:2111.13576* (2021).

[63] J Ben Schafer et al. "Collaborative filtering recommender systems". In: *The adaptive web*. Springer, 2007, pp. 291–324.

[64] Roan Schellingerhout, Volodymyr Medentsiy, and Maarten Marx. "Explainable Career Path Predictions using Neural Models". In: (2022).

[65] Tomoki Sekiguchi. "Person-organization fit and person-job fit in employee selection: A review of the literature". In: *Osaka keidai ronshu* 54.6 (2004), pp. 179–196.

[66] Xiao Sha, Zhu Sun, and Jie Zhang. "Hierarchical attentive knowledge graph embedding for personalized recommendation". In: *Electronic Commerce Research and Applications* 48 (2021), p. 101071.

[67] Amit Kumar Sharma, Sandeep Chaurasia, and Devesh Kumar Srivastava. "Sentimental short sentences classification by using CNN deep learning model with fine tuned Word2Vec". In: *Procedia Computer Science* 167 (2020), pp. 1139–1147.

[68] Rahul Shrivastava, Dilip Singh Sisodia, and Naresh Kumar Nagwani. "Utility optimization-based multi-stakeholder personalized recommendation system". In: *Data Technologies and Applications* ahead-of-print (2022).

[69] Maxwell Szymanski, Martijn Millecamp, and Katrien Verbert. "Visual, textual or hybrid: the effect of user expertise on different explanations". In: *26th International Conference on Intelligent User Interfaces*. 2021, pp. 109–119.

[70] Sho Takase and Shun Kiyono. "Lessons on parameter sharing across layers in transformers". In: *arXiv preprint arXiv:2104.06022* (2021).

[71] Ilaria Tiddi and Stefan Schlobach. "Knowledge graphs as tools for explainable machine learning: A survey". In: *Artificial Intelligence* 302 (2022), p. 103627.

[72] Nava Tintarev and Judith Masthoff. "A survey of explanations in recommender systems". In: *2007 IEEE 23rd international conference on data engineering workshop*. IEEE. 2007, pp. 801–810.

[73] Nava Tintarev and Judith Masthoff. "Designing and evaluating explanations for recommender systems". In: *Recommender systems handbook*. Springer, 2011, pp. 479–510.

[74] Chirayu Upadhyay et al. "Explainable Job-Posting Recommendations Using Knowledge Graphs and Named Entity Recognition". In: *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (2021), pp. 3291–3296. ISSN: 1062922X. DOI: 10.1109/SMC52423.2021.9658757.

[75] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[76] Connie R Wanberg, Abdifatah A Ali, and Borbala Csillag. "Job seeking: The process and experience of looking for a job". In: *Annual Review of Organizational Psychology and Organizational Behavior* 7 (2020), pp. 315–337.

[77] Xiang Wang et al. "Kgat: Knowledge graph attention network for recommendation". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 950–958.

[78] J.D. Werbel and S.W. Gilliland. "Person-environment fit in the selection process". In: *G. R. Ferris (Ed.), Research in human resources management* 17 (), pp. 109–243.

[79] Sarah Wiegreffe and Yuval Pinter. "Attention is not not explanation". In: *arXiv preprint arXiv:1908.04626* (2019).

[80] Haibing Wu et al. "Aspect-based opinion summarization with convolutional neural networks". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2016, pp. 3157–3163.

[81] Wenpeng Yin et al. "Comparative study of CNN and RNN for natural language processing". In: *arXiv preprint arXiv:1702.01923* (2017).

[82] Ezgi Yıldırım, Payam Azad, and Şule Gündüz Öğüdücü. "biDeepFM: A multi-objective deep factorization machine for reciprocal recommendation". In: *Engineering Science and Technology, an International Journal* 24.6 (2021), pp. 1467–1477.

[83] Hang Zhang et al. "Poolingformer: Long document modeling with pooling attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12437–12446.

[84] Lujun Zhao et al. "Sequence labeling with deep gated dual path CNN". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.12 (2019), pp. 2326–2335.

[85] Chen Zhu et al. "Person-job fit: Adapting the right talent for the right job with joint representation learning". In: *ACM Transactions on Management Information Systems (TMIS)* 9.3 (2018), pp. 1–17.

[86] Barret Zoph et al. "Designing effective sparse expert models". In: *arXiv preprint arXiv:2202.08906* (2022).