JTH: A Dataset for Evaluating Cold-Start and Temporal Dynamics in Job Recommendation

Yann Millet
yann.millet@telecom-sudparis.eu
Télécom SudParis, IPParis, Samovar
Palaiseau, France

Éric Behar
eric.behar@telecom-sudparis.eu
Télécom SudParis, IPParis, Samovar
Palaiseau, France

Julien Romero
julien.romero@telecom-sudparis.eu
Télécom SudParis, IPParis, Samovar
Palaiseau, France

ABSTRACT

Job-matching platforms operate under brief and asymmetric lifetimes as vacancies close in weeks while candidates surface for only a few days. So, real-world recommenders must cope simultaneously with pervasive cold start, extreme sparsity, and shifting temporal overlap. Public benchmarks, however, lack rich two-sided semantics and fine-grained time stamps, making such research impossible. We present **Job Tracking History (JTH)**, a seven-year corpus curated by professional recruiters that pairs structured profiles for 38k candidates and 6k vacancies with day-level traces for 43k application trajectories, covering every stage from shortlist to offer. All data are de-identified via k-anonymity, noise on numeric fields, and coarse geocoding. We detail the collection and cleaning pipeline, report descriptive statistics that expose JTH's heavy-tailed lifetimes and dense skill information, and release (i) the anonymized tables, (ii) a time-respecting evaluation script for both candidate and job cold-start scenarios, and (iii) baseline implementations that blend semantic, temporal and collaborative signals, establishing a new test bed for research on fast-moving, semantically rich job recommendation.

CCS CONCEPTS

• **Computing methodologies** → *Semantic networks*; *Learning to rank*; *Neural networks.*

KEYWORDS

dataset, recommender system, temporality, job recommendation, cold start

ACM Reference Format:
Yann Millet, Éric Behar, and Julien Romero. 2018. JTH: A Dataset for Evaluating Cold-Start and Temporal Dynamics in Job Recommendation. In *Proceedings of RecSys '25: ACM Conference on Recommender Systems (RecSys '25).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Online job-matching platforms face a fundamentally different landscape from e-commerce or media recommendation. A job posting is typically live for only a few weeks, while an active candidate disappears as soon as they accept an offer. Besides, both sides may enter and leave the market multiple times over a career. These fleeting lifetimes (median 25 days for jobs, 10 days for candidates in our data) yield three compounding hurdles: (i) pervasive **cold start** [6, 11, 21, 32, 38] for every newly posted vacancy and every first-time applicant, (ii) extreme **interaction sparsity**, and (iii) **complex temporal overlap**, because the window during which two entities can actually meet is narrow and highly variable (the median application process is of 8 days in our data).

	Statistics
Number of candidates	37,754
Number of jobs	6,095
Number of application processes	43,294
Number of candidate features	12
Number of job features	11
Number of application steps	11
Timespan	03/2018-04/2025

Table 1: Core Statistics

Although the recommendation community has made great progress on time-aware prediction in sequential datasets such as MovieLens [17], Netflix [5], and Gowalla [9], and on session-based tasks such as the RecSys Challenge'15 [4], those corpora differ from job-matching in two decisive ways. First, users in entertainment or shopping sites produce dense action logs that provide context for learning user states; job seekers do not. Second, items in those domains have long or indefinite lifespans, so temporal modeling focuses on recurrent consumption patterns rather than entity birth and death. Public job-oriented datasets (e.g., XING RecSys'17 [1], CareerBuilder [7]) partially acknowledge cold start but provide little fine-grained timing and almost no rich semantic descriptors of both the job and the candidate sides. Generic temporal corpora such as Yelp [8] or Gowalla suffer from the same limitation: they capture when interactions occur but not who the participants are at a level that would enable feature-based cold-start reasoning.

Yet, the recruiting sector is in dire need of automated processes [30], and many works have proposed to build recommender systems to assist recruiters [12, 14, 25, 26, 33, 36]. However, most of them rely on heuristics based on manual features and trained models on very noisy click data. Recently, some approaches started using large language models [33] to better understand job postings and resumes, but with limited results due to the difficulty of considering the collaborative filtering signals.

RecSys '25, Sep. 22–26, 2025, Prague, Czech Republic

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

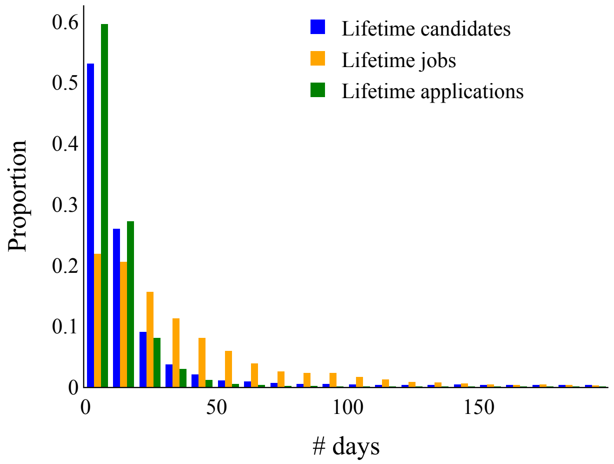

Figure 1: Lifetime of the candidates, jobs, and application processes

We introduce Job Tracking History (JTH) to close this gap. Collected and curated by professional recruiters over seven years, JTH combines breadth with depth:

- 37,754 candidates, 6,095 jobs, 43,294 interaction events
- Rich semantic context for candidates (creation date, skills, expertise area, job category, years of experience, salary, contract type, location, source) and for jobs (creation date, job category, skills, contract type, expertise area, years of experience, location, start date, salary)
- Multi-stage interaction traces, from recruiter pre-selection through interview, offer, and final acceptance, making it possible to frame tasks as coarse binary classification or as fine-grained stage prediction
- All records are fully de-identified through a multi-stage anonymization pipeline; licensing and privacy details accompany the release

This paper makes three contributions:

- **Dataset**. We document the collection, cleaning, anonymization, and temporal splitting of JTH, and provide descriptive statistics that expose its heavy-tailed lifetimes and the short application trajectories
- **Evaluation framework**. We define reproducible protocols for candidate cold start, job cold start, and rolling time-slice recommendation, along with clearly separated train/test windows.
- **Baselines and metrics** We benchmark representative collaborative filtering, heuristic, and trained models using standard metrics (P@k, MRR, NDCG@K), establishing reference scores.

By coupling high-resolution temporal signals with rich semantic context on both sides of the market, JTH enables a systematic study of exactly the phenomena that hamper real-world job-matching systems but remain under-explored in existing benchmarks. We release the anonymized data, code, and leaderboard scripts to foster research on temporally aware, semantically grounded job recommendations.

2 PREVIOUS WORKS

2.1 Public Benchmarks and Their Limits

General-purpose recommendation corpora. The long-standing MovieLens [17] series and the Netflix Prize [5] data have driven most algorithmic progress, but both expose semantics only on the item side (film genre, year) while users remain anonymous IDs with timestamps. Review-centric sets such as Amazon [18] and Yelp [8] add rich textual product or venue descriptions yet still provide virtually no structured user attributes, and items seldom expire, so real cold-start pressure and entity churn are absent.

Sequential click-stream datasets. Session challenges such as YooChoose (RecSys '15) [4] capture second-scale action sequences, supporting research on recurrent neural networks and Transformer [31] recommenders, but sessions last minutes and the same catalogue lives throughout the timeline, leaving birth-and-death dynamics unexplored.

Job-matching datasets. Open corpora for recruitment are rare. The Kaggle CareerBuilder challenge (2012) [7] shipped candidates and jobs. Candidate rows carried city, state, country, ZIP, degree type, major and graduation date, while jobs included title, description, requirements and dates. Interactions were single "application" events derived from click logs, with no multi-stage timestamps. The XING RecSys-17 [1] dataset addressed cold start but the license lapsed after the competition, making the data inaccessible and hard to compare with subsequent work.

Across these resources, three gaps persist:

- *Dual-sided semantics.* Item metadata abound but user-side structure is missing or extremely coarse.
- *Fine-grained time.* Only posting and click dates are available; there is no trace of the recruiting funnel.
- *Continual cold start.* Entities live for months or years, so algorithms are never forced to operate with one-shot histories.

2.2 Models That Exploit Semantics and Time

Content-based and hybrid approaches, ranging from skill-overlap heuristics [22, 24] to deep joint encoders [20, 37] or knowledge-graph embeddings [2, 3, 16], leverage item text and attributes to mitigate cold start, while TimeSVD++ [19] and its neural descendants [10, 34, 35] model temporal drift in user preferences. Such models show strong gains on static domains, but when we applied representative versions to our markedly sparser, fast-churning data they over-fitted and under-performed simpler online baselines; we therefore concentrated the experiments on interpretable models that separately highlight semantic, temporal and collaborative signals.

2.3 How JTH Advances the State of Resources

JTH is, to our knowledge, the first publicly distributable corpus that combines:

- rich structured features on both sides (skills, contract types, locations, seniority),
- day-level timestamps for every stage of 43k recruiting trajectories, and

- natural, frequent cold-start events born of median lifetimes of only one to two weeks.

These characteristics enable systematic study of algorithms that must jointly reason over semantics, collaborative feedback and rapid temporal dynamics, an experimental scenario that existing datasets cannot support.

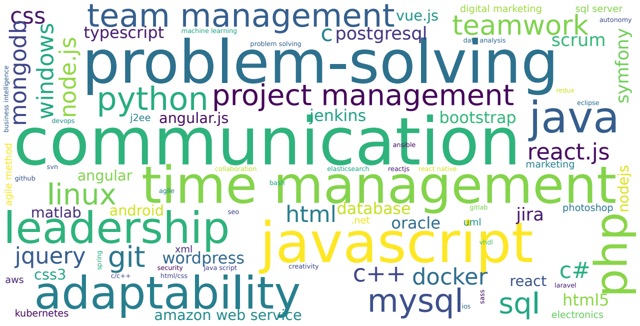

Figure 2: Wordcloud of the skills of the candidates

3 THE JTH DATASET

3.1 Data Collection

JTH was assembled in-house by a France-based recruiting firm that logged every action in its applicant-tracking system (ATS) from 2018 to 2025. About ten professional headhunters sourced vacancies directly from client companies and drew candidates either from professional networks (principally LinkedIn) or via the firm's web portal. For each job–candidate pair, they recorded a canonical sequence of events: pre-selection, headhunter interview, resume submission to the company, employer-side interviews (when feedback was returned), and final offer, each stamped with a time. Throughout the process, recruiters manually filled a common schema: candidates carry creation date, skills, expertise area, job category, years of experience, salary or daily rate, contract type, geographic area, and source; jobs mirror most of these fields. Locations are stored at city or region granularity.

3.2 Data Content

JTH is released as three UTF-8, comma-separated tables that can be joined through surrogate keys. *candidates.csv* contains one row per applicant and the columns candidate id (primary key), create date, and nine profile attributes (skills, expertise area, job category, years of experience, actual salary, actual daily salary, contract type, location, source). *jobs.csv* mirrors this structure with job id as the key and analogous descriptive fields for each vacancy. Multi-valued attributes such as skills and, occasionally, contract type are stored as semicolon-delimited strings.

The interaction trace lives in *history.csv*. Each record is uniquely identified by the pair *(candidate id, job id)* and contains a fine-grained timeline of the recruitment funnel (Figure 3): application entry date, shortlist date, qualification date, resume sent to company date, up to four interview timestamps (1st interview date, ..., 4th interview date), job offer proposed date, job offer accepted date, and a possible end of process date for negative outcomes. These tables together provide entity-level context and event-level chronology for every job–candidate encounter in the dataset.

3.3 Data Cleaning

Numerous fields were hand-entered and, therefore, noisy. All timestamps were converted to day-level resolution (YYYY-MM-DD). Salaries, always denominated in euros, appeared as "45 k €", "45,000", "45k gross/yr", etc.; we stripped punctuation, symbols, and free text, then stored a numeric annual-gross value. Years of experience were mapped to four bins: 0–2 years, 3–6 years, more than 6 years, and unknown. French categorical labels for job category, contract type, and expertise area were translated through a manual lookup table. Postcodes were normalized to the five-digit INSEE standard after removing non-numeric characters.

Unstructured text required additional processing. We ran LLaMa 3.3 80 B [15] in extraction mode over every resume and job description, capturing plain-string lists of (i) technical skills, (ii) soft skills, and (iii) spoken languages. Then, the textual resumes and job descriptions are removed to help with anonymization (see Section 3.4).

Finally, the interaction log was checked for temporal consistency, e.g., an interview cannot precede resume submission. Fewer than 200 candidate–job traces were dropped outright, and roughly 2,500 implausible event stamps were removed; the remaining sequences respect the canonical order and are ready for downstream modeling.

3.4 Data Anonymization

The raw corpus comprised three relational tables. All primary keys (candidate id, job id, application id) were replaced by deterministic digests, preserving referential integrity while meeting the GDPR definition of pseudonymized data [13]. Direct personal fields (names, free-text resumes, contact details, company names, demographic attributes, job descriptions) were removed outright.

Numeric attributes. Salaries and daily rates were first smoothed by micro-aggregation [28] and then perturbed with Laplace noise [23] (privacy parameter $\epsilon = 3$), before rounding to whole euros. All creation and application timestamps received zero-mean Laplace noise at day granularity; values were then clipped so that every recruitment trajectory retained its correct internal order.

Quasi-identifiers. Geography was coarsened to the French département code (two digits max). Years of experience were bucketed into broad seniority bands. List-style fields (skills, expertise area, job category) were tokenized, lower-cased, deduplicated, and frequency-filtered: infrequent tokens (less than $f = 20$ occurrences) were collapsed into the sentinel rare_skill.

k-Anonymity. We enforced k = 5 anonymity [29] jointly on the candidate and job tables over the quasi-identifier triple (zipcode, years of experience, contract type). Equivalence classes smaller than five were suppressed, and suppression cascaded to the interaction log to avoid dangling foreign keys. Finally, identifiers were re-indexed to close gaps.

Utility. Tables 3 and 4 compare each baseline's performance on the raw data (labelled "–Anon.") with its performance after

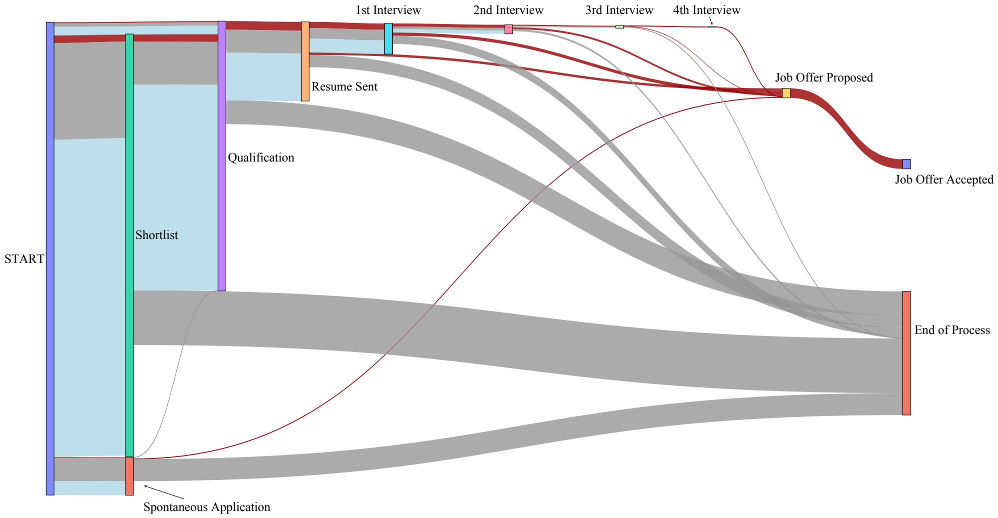

Figure 3: Sankey diagram of the recruiting process. The red lines represent successful recruitment. Grey lines represent explicit unsuccessful recruitment. Blue lines are unfinished application processes (often also negative).

anonymization. As anticipated, scores fall across the board, but the impact is uneven. The pure temporal baseline suffers most, its mean reciprocal rank (MRR) dropping by almost an order of magnitude, confirming that the day-level noise we inject blurs precise recency cues. By contrast, logistic regression, which blends many static signals, loses far less, while the recency-aware collaborative-filtering models retain virtually all of their accuracy. The latter result indicates that coarse time bins remain informative when they are combined with interaction structure, and, as discussed in Section 4.2, the added noise may even mitigate recruiter-specific biases buried in the raw timestamps.

The released dataset therefore satisfies k-anonymity, includes calibrated noise on sensitive numerics, and retains the relational structure required for realistic research on temporally aware job recommendation.

3.5 Data Statistics

JTH comprises 37,754 candidates, 6,095 jobs, and 43,294 candidate–job interaction traces (see Table 1). Entities are short-lived: candidates remain active a median of 10 days (mean 86 days) and vacancies 25 days (mean 43 days); the median application process lasts 8 days (see Figure 1).

Recruitment funnel. Figure 3 (Sankey) visualizes stage attrition. Numerically, 88% of pairs reach the shortlist, 56% pass the recruiter interview (qualification), but only 17% are forwarded to the hiring firm, and 2% culminate in an accepted offer; 28% terminate with an explicit rejection (KO). Spontaneous applications (8%) rarely progress: 89% end before a resume is sent.

Number of processes. As shown in Figures 4 and 5, candidates are often presented with very few opportunities, whereas recruiters try to provide enough candidates for a job posting. This stresses

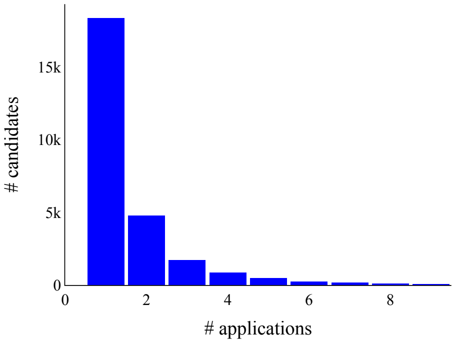

Figure 4: Number of applications associated with each candidate

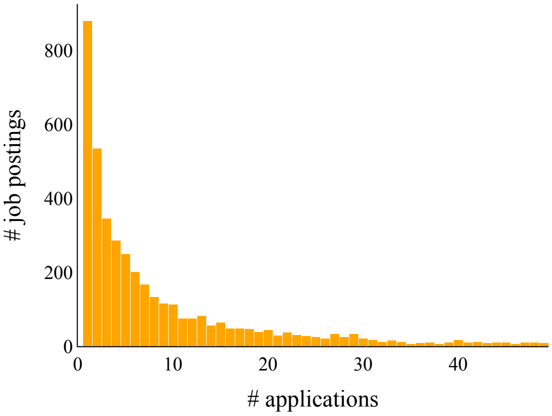

Figure 5: Number of applications associated with each job

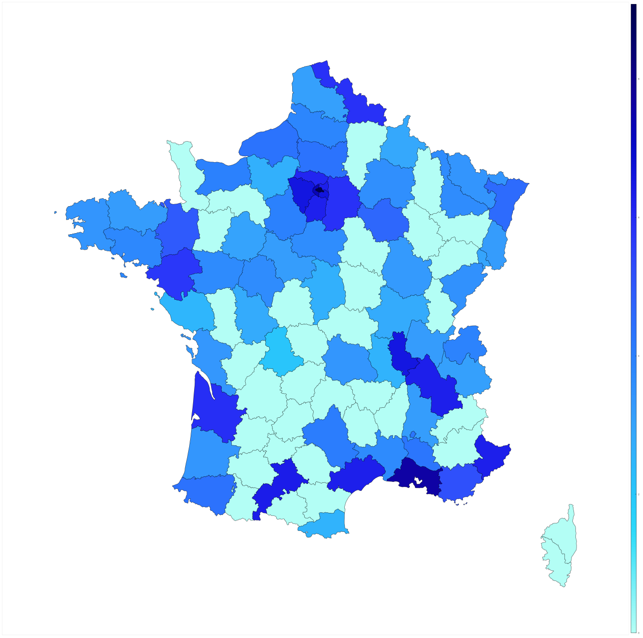

Figure 6: Geographic distribution of candidates, on a logarithmic scale

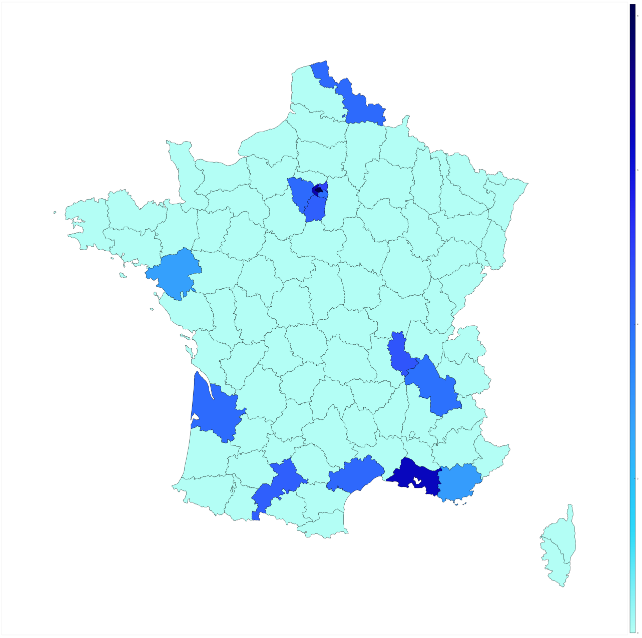

Figure 7: Geographic distribution of jobs, on a logarithmic scale

the difficulty of dealing with the cold start problem, in particular for candidates.

Profile mix. Among candidates with non-missing values, years of experience buckets are 0–2 years (1,030), 3–6 years (2,257), and more than 6 years (32,956). The most common expertise areas are Development (22k), Electronics (5k), and Functional IT (5k); contract types skew permanent (25k) over freelance (9k). Jobs show similar proportions, but many omit the years of experience field (only 690 filled).

Skill vocabulary. On average, a candidate is associated with 18 skills, and a job with 13. The ten most frequent technical skills are JavaScript (13k), PHP (12k), Python (11k), Java (11k), SQL (10k), Project Management (10k), Linux (9k), Git (9k), C++ (7k), C (6k), HTML (6k), and Docker (6k); on the job side, PHP, JavaScript and React dominate. See Figure 2 for a wordcloud of the candidates' skills.

Geolocation. The Figures 6 and 7 show the geographical distribution of candidates and jobs. Unsurprisingly, the jobs are mainly centered around large cities such as Paris, Lyon, Marseille, Bordeaux, Toulouse, and Montpellier. The distribution of candidates is more scattered.

Temporal distribution. Figures 8, 9, and 10 show the temporal distribution of candidates, jobs, and application processes. After a first warmup phase, the number of creations stabilizes and shows some seasonal effects as summer is less active in recruiting. We also observe no direct effect of COVID, showing that the recruiting industry managed to keep working thanks to remote work.

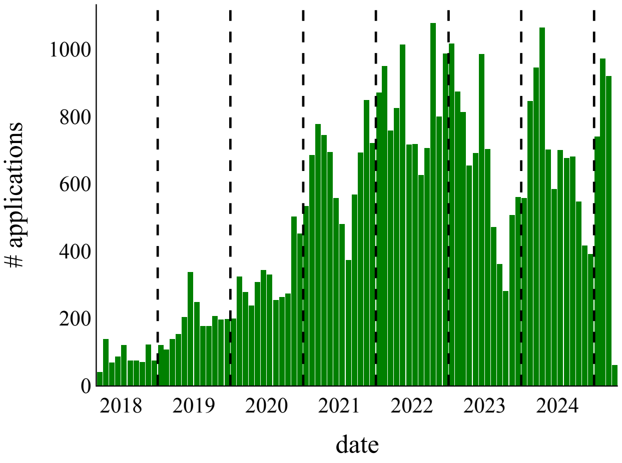

Figure 8: Timeline of the application processes

Missingness. As discussed previously, event-level sparsity is intrinsic to recruitment. Table 2 describes in detail the missing values for the features of the entities. As many fields are completed by hand, there are missing in many cases, especially for candidates. Some features were introduced more recently, making them sparser. We believe that some features can be inferred from the resume with appropriate training. This missingness naturally leads us to hybrid models, combining features with collaborative filtering signals and temporal information.

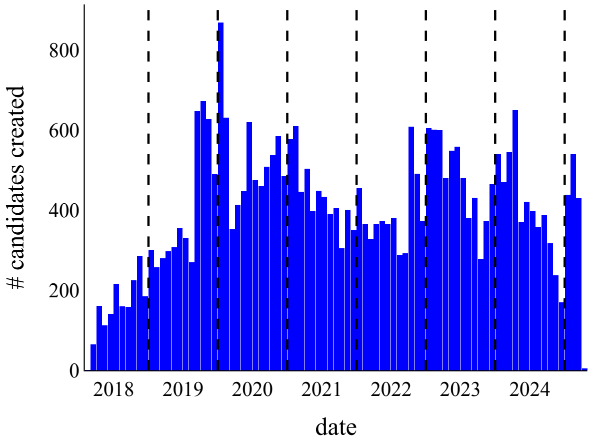

Figure 9: Timeline of candidates creations

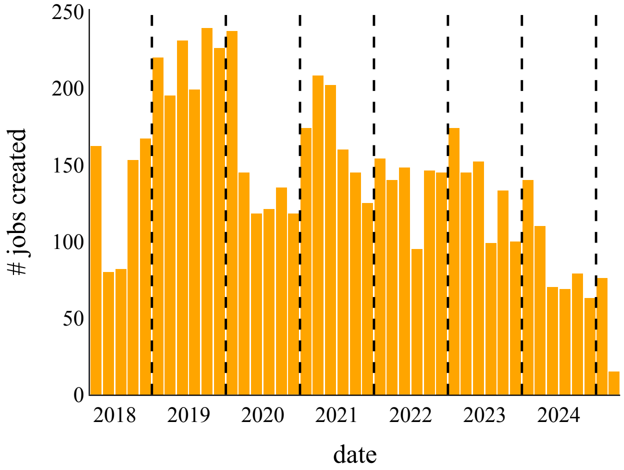

Figure 10: Timeline of job creations

	candidates	jobs
creation date	7%	0%
skills	5%	0%
expertise area	8%	30%
job category	27%	30%
years experience	83%	89%
salary	94%	34%
daily rate	98%	68%
contract	15%	0%
zipcode	41%	68%

Table 2: Missing values

These statistics underline the dual challenges JTH was built to address: **pervasive cold start** (little history per entity) and **extreme temporality** (fast churn and highly-skewed funnel progression).

3.6 Data Use

The JTH corpus is released under a CC BY-NC 4.0 license: it may be freely used for non-commercial research provided that the dataset is cited in any ensuing publication and the original license text is redistributed. Any attempt to re-identify individuals or companies, or to deploy models trained on JTH in a production recruiting service, is expressly forbidden.

The dataset is designed primarily for time-aware recommender-system research, with particular emphasis on dual cold-start scenarios and fast-moving temporal dynamics. Its rich attribute space (skills, expertise areas, contract types, geolocation) also enables graph-based methods such as skill co-occurrence networks, job–skill bipartite embeddings, and temporal knowledge-graph completion, as well as ancillary studies in representation learning, fairness auditing, churn prediction, and funnel analytics. Baseline implementations that accompany the release (see Section 4.2) are intended as starting points and carry the same license.

Ethics and privacy. All personal and corporate identifiers have been removed, yet residual risk remains; users must not combine JTH with external data to infer sensitive information. Furthermore, the corpus may contain latent biases. Although sex or ethnicity are not recorded, they can surface indirectly through correlated features; recruiter behavior introduces clustering effects among candidates, and strong temporal autocorrelation links entity creation dates with interaction likelihoods. Researchers should therefore audit models for both demographic and temporal bias before drawing substantive conclusions. A detailed discussion of these issues and bias-aware evaluation appears in Section 4.2.

4 EXPERIMENTS

4.1 Experimental Setup

Problem. The task is framed as time-conditioned ranking: given a timestamp T, rank candidates for a job ($c \longrightarrow j$) or jobs for a candidate ($j \longrightarrow c$) according to the probability that an interaction will occur at T. Training therefore reduces to binary classification. Observed pairs are positive, unobserved pairs are negative, and ranking scores are obtained from the predicted probabilities.

Temporal evaluation protocol. All splits respect chronological order. The reference script segments the data at the 80th percentile of the global timeline (train is from March 2018 to March 2024, test is from March 2024 to April 2025). Validation folds, when required, are carved from the tail of the training window. Because nearly every entity in the test set is new or has an extremely short history, the protocol naturally stresses cold-start performance. When required, for each positive in train we sample one random negative (ratio = 1:1 by default), although the scripts allow arbitrary ratios.

Metrics. We report ranking metrics, computed separately for $c \longrightarrow j$ and $j \longrightarrow c$ and then macro-averaged: MRR, Precision@K, Recall@K, F1@K, NDCG@K (Normalized Discounted Cumulative Gain) and MAP@K (Mean Average Precision) with $K \in \{1, 5, 10\}$. These metrics use the full candidate (or job) set.

Baselines. We implemented the following standard baselines:

- *Random.* Uniform ranking.

Table 3: Ranking metrics – Candidates → Jobs

Baseline	MRR	P@1	P@5	P@10	R@1	R@5	R@10	F1@1	F1@5	F1@10	NDCG@1	NDCG@5	NDCG@10	MAP@1	MAP@5	MAP@10
Random	0.002	0.000	0.000	0.000	0.000	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001
Jaccard	0.015	0.004	0.003	0.003	0.004	0.015	0.032	0.004	0.005	0.006	0.004	0.009	0.015	0.004	0.008	0.010
Popularity	0.004	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
Popularity + recency	0.085	0.030	0.022	0.018	0.030	0.110	0.177	0.030	0.037	0.032	0.030	0.070	0.091	0.030	0.057	0.066
Temporal	0.031	0.011	0.006	0.005	0.011	0.031	0.050	0.011	0.010	0.009	0.011	0.021	0.027	0.011	0.018	0.021
Past temporal	0.051	0.006	0.010	0.011	0.006	0.052	0.110	0.006	0.017	0.020	0.006	0.028	0.047	0.006	0.021	0.028
User-based CF	0.059	0.037	0.016	0.010	0.037	0.078	0.102	0.037	0.026	0.018	0.037	0.058	0.066	0.037	0.052	0.055
MF CF	0.033	0.021	0.008	0.005	0.021	0.041	0.050	0.021	0.014	0.009	0.021	0.032	0.035	0.021	0.028	0.030
CF + recency	0.122	0.081	0.028	0.017	0.081	0.138	0.173	0.081	0.046	0.032	0.081	0.111	0.122	0.081	0.102	0.107
Logistic reg.	**0.226**	**0.122**	**0.067**	**0.044**	**0.122**	**0.334**	**0.437**	**0.122**	**0.111**	**0.080**	**0.122**	**0.231**	**0.265**	**0.122**	**0.197**	**0.211**
Logistic reg. no time	0.019	0.004	0.004	0.005	0.004	0.020	0.050	0.004	0.007	0.009	0.004	0.011	0.021	0.004	0.008	0.012
Random forest	0.118	0.033	0.036	0.032	0.033	0.178	0.321	0.033	0.059	0.058	0.033	0.103	0.150	0.033	0.079	0.098
Random forest no time	0.020	0.003	0.005	0.005	0.003	0.025	0.053	0.003	0.008	0.010	0.003	0.013	0.022	0.003	0.009	0.013
Past temporal -Anon.	0.129	0.039	0.039	0.034	0.039	0.196	0.342	0.039	0.065	0.062	0.039	0.117	0.164	0.039	0.091	0.110
CF + recency -Anon.	0.073	0.034	0.018	0.012	0.034	0.092	0.121	0.034	0.031	0.022	0.034	0.063	0.073	0.034	0.054	0.058
Logistic reg. -Anon.	0.323	0.221	0.087	0.053	0.221	0.436	0.532	0.221	0.145	0.097	0.221	0.333	0.363	0.221	0.298	0.311

Table 4: Ranking metrics – Jobs → Candidates

Baseline	MRR	P@1	P@5	P@10	R@1	R@5	R@10	F1@1	F1@5	F1@10	NDCG@1	NDCG@5	NDCG@10	MAP@1	MAP@5	MAP@10
Random	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Jaccard	0.003	0.001	0.001	0.001	0.001	0.003	0.007	0.001	0.001	0.001	0.001	0.002	0.003	0.001	0.002	0.002
Popularity	0.006	0.002	0.002	0.001	0.002	0.008	0.011	0.002	0.003	0.002	0.002	0.006	0.006	0.002	0.005	0.005
Popularity + Recency	0.015	0.007	0.004	0.003	0.007	0.019	0.028	0.007	0.006	0.005	0.007	0.014	0.016	0.007	0.012	0.013
Temporal	0.008	0.003	0.002	0.002	0.003	0.010	0.016	0.003	0.003	0.003	0.003	0.006	0.008	0.003	0.005	0.006
Past temporal	0.019	0.000	0.001	0.002	0.000	0.003	0.021	0.000	0.001	0.004	0.000	0.001	0.007	0.000	0.001	0.003
User-based CF	0.014	0.006	0.004	0.003	0.006	0.019	0.031	0.006	0.006	0.006	0.006	0.012	0.016	0.006	0.010	0.012
MF CF	0.023	0.017	0.006	0.003	0.017	0.028	0.033	0.017	0.009	0.006	0.017	0.022	0.024	0.017	0.021	0.021
CF + recency	0.011	0.003	0.002	0.002	0.003	0.011	0.019	0.003	0.004	0.003	0.003	0.007	0.010	0.003	0.006	0.007
Logistic reg.	**0.039**	**0.018**	**0.011**	0.008	**0.018**	**0.053**	0.081	**0.018**	**0.018**	0.015	**0.018**	**0.035**	**0.045**	**0.018**	**0.030**	**0.033**
Logistic reg. no time	0.007	0.001	0.002	0.001	0.001	0.008	0.013	0.001	0.003	0.002	0.001	0.005	0.006	0.001	0.004	0.004
Random forest	**0.039**	0.011	0.010	**0.009**	0.011	0.048	**0.091**	0.011	0.016	**0.017**	0.011	0.029	0.043	0.011	0.023	0.029
Random forest no time	0.004	0.000	0.001	0.001	0.000	0.005	0.010	0.000	0.002	0.002	0.000	0.003	0.004	0.000	0.002	0.002
Past temporal -Anon	0.124	0.043	0.037	0.032	0.043	0.184	0.320	0.043	0.061	0.058	0.043	0.114	0.157	0.043	0.091	0.108
CF + recency -Anon.	0.008	0.002	0.001	0.001	0.002	0.007	0.011	0.002	0.002	0.002	0.002	0.005	0.006	0.002	0.004	0.004
Logistic reg. -Anon.	0.085	0.054	0.023	0.014	0.054	0.113	0.143	0.054	0.038	0.026	0.054	0.085	0.095	0.054	0.076	0.080

- *Temporal / Past-temporal*. Sort by | creation date − T |, with or without future entities.
- *Popularity (online) and Popularity + Recency*. Cumulative or exponentially decayed interaction counts.
- *Online user-based collaborative filtering (CF) and User-CF + Recency*. Cosine similarity over the live interaction matrix, updated daily; the recency variant ignores entities older than 90 days.
- *Online matrix-factorisation CF* Incremental SGD on a latent factor model.
- *Skill Jaccard* The score is the Jaccard similarity between the candidate and job skill sets.
- *Online feature-based classifiers*. Logistic regression and random forest are trained on boolean matches of skills, contract type, expertise area, job category, years-experience bucket, and temporal distances; models are refreshed each day with new data.

Runtime environment All code is Python 3.11 and relies only on open-source libraries (scikit-learn, Pandas). Experiments were executed on a commodity laptop (Intel i7, 32 GB RAM) in under 30 minutes end-to-end; the online feature-based models were the slowest components.

4.2 Results

We structure the discussion around four questions that reflect the design goals of JTH. All numbers refer to the chronologically held-out test window and use MRR as the headline indicator; NDCG@10 and the other rank-oriented metrics follow the same trends. The results are presented in Figures 3 and 4

RQ 1: Is it intrinsically easier to recommend jobs to candidates than candidates to jobs? Yes. Every run scores markedly higher when the system proposes vacancies to an active job-seeker (*candidates → jobs*) than when it proposes applicants to an open vacancy (*jobs → candidates*). For example, one of the strongest models we tried, a random forest fed with semantic and temporal features, reaches an MRR of 0.118 in the *candidates → jobs* direction but only 0.039 in the reverse orientation. Logistic regression shows the same three-to-one ratio (0.226 vs. 0.040). The gap highlights the asymmetry built into professional recruiting: many seekers are ready to accept one among several similar roles, whereas employers often require a near-perfect match that may never have appeared in the system before. Another reason is that there are many more candidates than jobs, making the recommendation of candidates harder.

RQ 2: What do we gain from pure semantic matching? To isolate semantics, we compare the Jaccard skill overlap with a logistic-regression model that uses only static profile features and excludes every temporal signal. In the *candidates → jobs* task, Jaccard reaches an MRR of 0.015 and the logistic model only nudges that to 0.019; in the harder *jobs → candidates* direction, the two methods sit at 0.003 and 0.007 respectively. The narrow gap shows that static content alone is a weak predictor under extreme cold start and sparsity. The information is present (otherwise the later experiments could not succeed) but it requires more sophisticated modeling, conditioning on time, and interactions to become truly useful.

RQ 3: How much do collaborative-filtering signals contribute? When we move from literal skill overlap to online user-based CF, the *candidates → jobs* MRR quadruples to 0.059, and the *jobs → candidates* score climbs to 0.014. A matrix-factorization variant performs similarly (0.033 / 0.023). These gains confirm that even sparse interaction histories carry patterns that generic similarity measures cannot capture. Yet CF by itself still falls short of models that additionally look at content or time, underscoring the need to blend signals. It is important to notice that there might be biases added by the recruiters: they often know very well some candidates, and tend to group them to suggest them to similar candidates. This grouping behavior helps with collaborative filtering.

RQ 4: How critical is temporal information? The effect of recency is dramatic. Simply weighting popularity counts by an exponential decay pushes the popularity baseline from essentially unusable (MRR of 0.004–0.006) to 0.085 in *candidates → jobs* and 0.015 in *jobs → candidates*, a twenty-fold jump in the easier task. Adding a temporal filtering to user-based CF doubles its *candidates → jobs* performance (0.059 to 0.122), although it hurts slightly in the tougher *jobs → candidates* setting where overlap is already scarce. The decisive role of time is clearest in the learned models: removing the timestamp features from the random forest collapses its *candidates → jobs* MRR from 0.118 to 0.020, and stripping them from logistic regression drags the score from 0.226 to 0.019. In other words, without temporal context, the rich semantics are almost useless. However, we need to be conscious that there are some biases due to the recruiting process: a recruiter finds a position to fill, then gathers relevant candidates and suggests them to the company. If nothing works, the recruiter might perform a second wave of suggestions. Therefore, the temporal data follows this temporal behavior of the recruiters, which might be very different from spontaneous applications on platforms such as LinkedIn.

Summary. The experiments converge on three lessons. **First, cold-start recruiting demands temporal awareness; freshness trumps all other cues. Second, interaction structure carries an additional signal that pure content matching misses. Third, semantics matter, but only when the model is able to align them with the clock and with observed behavior.** These observations validate the design of JTH, whose fine-grained timestamps, rich profiles, and preserved interaction graph together create a challenging benchmark that cannot be solved by any single modality alone.

5 CONCLUSION

This paper introduced JTH, a five-year corpus of 38k candidates, 6k vacancies, and 43k interaction trajectories curated by professional headhunters. JTH brings together three ingredients absent from existing benchmarks:

(1) Two-sided semantics: structured skills, seniority, contract types, and locations for both candidates and jobs;
(2) Day-level temporal traces that cover the entire recruitment funnel, from shortlist to final offer;
(3) Natural cold start driven by the short life-spans of entities (median 10–25 days).

We documented the end-to-end pipeline (data collection, cleaning, anonymization) and supplied a time-respecting evaluation script plus a suite of online baselines. Experiments showed that none of the individual signals (content, interactions, recency) suffices on its own; useful performance arises only when models combine semantics, collaborative structure and strong temporal priors, validating the design choices behind the dataset.

Limitations. JTH is France-centric and omits demographic attributes, so studies on geographic transfer or fairness across gender and ethnicity remain out of scope. Skill lists were extracted with a large language model and inevitably contain noise, although manual spot-checks suggest higher coverage than alternative parsers. The interaction log is modest in size but high in annotation quality; several profile fields are hand-entered, which may not exist in other ATS platforms and could introduce recruiter-specific bias. Finally, despite aggressive k-anonymity and noise injection, residual privacy risk can never be reduced to zero when free text is involved.

Future directions. With rich semantics and clean temporal splits in place, the next research steps are methodological:

(1) *Funnel-aware ranking*. Weighting training events by the stage reached (resume sent, interview, offer) may align optimization more closely with business value than a flat binary label.
(2) *Heterogeneous graph learning*. JTH's attribute diversity enables models that jointly embed candidates, jobs, skills, and contract types; external vocabularies such as ESCO [27] can further regularize the skill graph.
(3) *Sequence and survival modeling*. Predicting time-to-event, or the likelihood that a vacancy expires without hire, could complement ranking.
(4) *Bias diagnostics*. Systematic audits for recruiter-dependency or temporal leakage will be important as more sophisticated models appear.

The code base (GitHub, https://github.com/Aunsiels/JTH) and the anonymized dataset (will be released to a truly public repository once accepted, https://partage.imt.fr/index.php/s/dcNDrS33NoKZLQR) will be released under CC BY-NC 4.0, providing a common ground for the community to explore these avenues while preserving privacy to the fullest extent possible.

REFERENCES

[1] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. 2017. Recsys challenge 2017: Offline and online evaluation. In *Proceedings of the eleventh acm conference on recommender systems*. 372–373.

[2] Eric Behar, Julien Romero, Amel Bouzeghoub, and Katarzyna Wegrzyn-Wolska. 2023. Tackling cold start for Job recommendation with heterogeneous graphs. In *CEUR Workshop Proceedings*, Vol. 3490.

[3] Eric Behar, Julien Romero, Amel Bouzeghoub, and Katarzyna Wegrzyn-Wolska. 2024. TIMBRE: Efficient Job Recommendation On Heterogeneous Graphs For Professional Recruiters. *arXiv preprint arXiv:2411.15146* (2024).

[4] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. 2015. Recsys challenge 2015 and the yoochoose dataset. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 357–358.

[5] James Bennett and Stan Lanning. 2007. The netflix prize. (2007).

[6] Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, and Changsheng Xu. 2023. User cold-start recommendation via inductive heterogeneous graph neural network. *ACM Transactions on Information Systems* 41, 3 (2023), 1–27.

[7] CareerBuilder. 2012. CareerBuilder's Job Recommendation Challenge. https://www.kaggle.com/competitions/job-recommendation [Accessed: 06-05-2025].

[8] CareerBuilder. 2019. CareerBuilder's Job Recommendation Challenge. https://business.yelp.com/data/resources/open-dataset/ [Accessed: 06-05-2025].

[9] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[10] Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. 2013. Modeling temporal adoptions using dynamic matrix factorization. In *2013 IEEE 13th international conference on data mining*. IEEE, 91–100.

[11] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-Augmented Meta-Optimization for Cold-start Recommendation. arXiv:2007.03183 [cs.IR]

[12] Mauricio Noris Freire and Leandro Nunes de Castro. 2021. e-Recruitment recommender systems: a systematic review. *Knowledge and Information Systems* (2021).

[13] EU GDPR. 2018. General data protection regulation (gdpr).

[14] Anna Giabelli, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Andrea Seveso. 2021. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing* 101 (2021), 107049.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[16] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3549–3568.

[17] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[18] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[19] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.

[20] Miklas S Kristoffersen, Jacob L Wieland, Sven E Shepstone, Zheng-Hua Tan, and Vinoba Vinayagamoorthy. 2019. Deep joint embeddings of context and content for recommendation. *arXiv preprint arXiv:1909.06076* (2019).

[21] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 1563–1573. https://doi.org/10.1145/3394486.3403207

[22] Abhinav Maurya and Rahul Telang. 2017. Bayesian multi-view models for member-job matching and personalized skill recommendations. In *2017 IEEE International Conference on Big Data (Big Data)*. 1193–1202. https://doi.org/10.1109/BigData.2017.8258045

[23] Iyiola E Olatunji, Jens Rauch, Matthias Katzensteiner, and Megha Khosla. 2024. A review of anonymization for healthcare data. *Big data* 12, 6 (2024), 538–555.

[24] Bharat Patel, Varun Kakuste, and Magdalini Eirinaki. 2017. CaPaR: a career path recommendation framework. In *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 23–30.

[25] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. 2020. An enhanced neural network approach to person-job fit in talent recruitment. *TOIS* (2020).

[26] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, and Rocky Bhatia. 2020. A Machine Learning approach for automation of Resume Recommendation system. *Procedia Computer Science* 167 (2020), 2318–2327.

[27] Johannes De Smedt, Martin le Vrang, and Agis Papantoniou. 2015. ESCO: Towards a Semantic Web for the European Labor Market. In *LDOW@WWW*. https://api.semanticscholar.org/CorpusID:14184714

[28] Agusti Solanas, Francesc Sebé, and Josep Domingo-Ferrer. 2008. Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*. 61–69.

[29] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.

[30] TalentWorks. 2019. Science of the job search. https://web.archive.org/web/20190322214104/http://talent.works/blog/category/science-of-the-job-search.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. arXiv:2107.05315 [cs.IR]

[33] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2023. Exploring Large Language Model for Graph Data Understanding in Online Job Recommendations. arXiv:2307.05722 [cs.AI]

[34] Ting Wu, Yong Feng, JiaXing Sang, BaoHua Qiang, and YaNan Wang. 2018. A novel recommendation algorithm incorporating temporal dynamics, reviews and item correlation. *IEICE transactions on Information and Systems* 101, 8 (2018), 2027–2034.

[35] Liang Xiang and Qing Yang. 2009. Time-dependent models in collaborative filtering based recommender system. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, 450–457.

[36] Chen Yang, Yupeng Hou, Yang Song, Tao Zhang, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Modeling Two-Way Selection Preference for Person-Job Fit. In *RecSys*.

[37] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*. 425–434.

[38] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (, Virtual Event, Canada,) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1167–1176. https://doi.org/10.1145/3404835.3462843