ConFit v2: Improving Resume-Job Matching using Hypothetical Resume Embedding and Runner-Up Hard-Negative Mining

Xiao Yu[*♠] **Ruize Xu**[*♠] **Chengyuan Xue**[*♦] **Jinzhong Zhang**[♣] **Xu Ma**[♣] **Zhou Yu**[♠♥]

[♠]Columbia University [♦]University of Toronto [♣]Intellipro Group Inc. [♥]Arklex.ai

{xy2437,zy2461}@columbia.edu

{jinzhong}@intelliprogroup.com

Abstract

A reliable resume-job matching system helps a company recommend suitable candidates from a pool of resumes and helps a job seeker find relevant jobs from a list of job posts. However, since job seekers apply only to a few jobs, interaction labels in resume-job datasets are sparse. We introduce ConFit v2, an improvement over ConFit to tackle this sparsity problem. We propose two techniques to enhance the encoder's contrastive training process: augmenting job data with hypothetical reference resume generated by a large language model; and creating high-quality hard negatives from unlabeled resume/job pairs using a novel hard-negative mining strategy. We evaluate ConFit v2 on two real-world datasets and demonstrate that it outperforms ConFit and prior methods (including BM25 and OpenAI text-embedding-003), achieving an average *absolute* improvement of 13.8% in recall and 17.5% in nDCG across job-ranking and resume-ranking tasks.

1 Introduction

Online recruitment platforms like LinkedIn serve over 990 million users and 65 million businesses, processing more than 100 million job applications each month (Iqbal, 2025). As these platforms continue to grow, there is a rising need for *efficient* and *robust* resume-job matching systems. A practical system that reliably identifies suitable candidates/jobs from large pools will save considerable effort for both employers and job seekers.

Since both resume and job posts are often stored as text data, there has been an increased interest in using transformer models to model resume-job fit (or referred to as "person-job fit"). Many prior works (Zhu et al., 2018; Qin et al., 2018; Bian et al., 2020; Yang et al., 2022; Shao et al., 2023) focus on designing complex modeling techniques to model resume-job matching. For example, APJFNN (Qin et al., 2018) uses hierarchical recurrent neural networks to process the job and resume content, and DPGNN (Yang et al., 2022) uses a dual-perspective graph neural network to model the relationship between resumes and jobs. Unlike these manual model/feature engineering methods, more recently Yu et al. (2024b) shows that dense retrieval techniques like contrastive learning achieve significant improvements without relying on any complex designs. By introducing dense retrieval methods to resume-job matching, ConFit (Yu et al., 2024b) presented a simple yet effective method to rank thousands of resumes/jobs in milliseconds, by combining training neural encoders such as E5 (Wang et al., 2022) with inner-product search algorithms such as FAISS (Johnson et al., 2019).

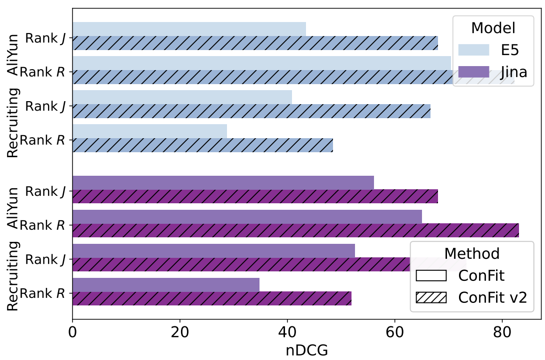

Figure 1: Performance comparison between ConFit v2 and ConFit across the AliYun and the Recruiting dataset. "Rank R" indicates ranking resume, and "Rank J" indicates ranking job.

We propose ConFit v2, an enhanced baseline that achieves an average absolute improvement of 13.8% in recall and 17.5% in nDCG on ranking resumes and jobs compared to ConFit (see Figure 1). We introduce two key improvements to enhance the neural encoder's training process: 1) Hypothetical Resume Embedding (HyRe) that leverages a large language model (LLM) to generate a hypothetical resume and augment the job post

[*]denotes equal contribution

Findings of the Association for Computational Linguistics: ACL 2025, pages 12775–12790
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

(see Figure 2), providing implicit details to reduce the burden on encoder training; and 2) Runner-Up Mining (RUM) that mines a large set of high-quality hard negatives to help the encoder better discern positive with near-positive samples. We train ConFit v2 using HyRe and RUM on two real-world datasets, and show that ConFit v2 achieves new state-of-the-art performance on ranking resumes and jobs, outperforming ConFit and other prior work (including BM25 and OpenAI text-embedding-003-large). We will open-source our code and data (under license agreements) to provide a strong baseline for future research in empowering resume-job matching systems with dense retrieval techniques.

2 Background

A resume-job matching (or often called *person-job fit*) system models the compatibility between a resume R and a job post J, allowing the systems to recommend the most suitable candidates for a given job, or suggest the most relevant jobs for a given candidate (Bian et al., 2020; Yang et al., 2022; Shao et al., 2023; Yu et al., 2024b). Since resumes and job posts are often stored as text data, many prior works consider using neural networks (e.g., encoders) to quantify the compatibility between resumes and jobs:

$$\text{score}(R, J) = f_\theta(R, J) \to \mathbb{R},$$

where f_θ could be directly modeled by a neural network (Zhu et al., 2018; Yang et al., 2022; Shao et al., 2023), or by computing inner product/cosine similarity (Yu et al., 2024b) between a resume embedding $f_\theta(R)$ and a job embedding $f_\theta(J)$.

Despite the rapid growth of online recruitment platforms and the increasing availability of job posts and resume data, there are *very few interaction labels* between any resumes and job posts (Bian et al., 2020; Yu et al., 2024b). This is because a candidate usually applies to only a few positions, and a job interviewer often reviews only a few resumes for a given job post. Often, the resulting dataset $\mathcal{D} = \{R_i, J_i, y_i\}$ has size $|\mathcal{D}| \ll n_R \times n_J$, where n_R and n_J are the total number of resumes and jobs respectively, and $y_i \in \{0, 1\}$ is a *binary* signal representing whether a resume R_i is short listed for a job J_i. For example, in both the Recruiting and AliYun datasets (Section 4.1) used in this work, **less than 0.05%** of the total possible (resume, job) pairs are annotated. This label sparsity poses challenges in: 1) crafting high-quality training data/hard negatives for training a neural encoder f_θ; and 2) learning a representation space that generalizes well across diverse resumes and job posts.

3 Approach

We propose ConFit v2, a simple and general-purpose approach that improves ConFit (Yu et al., 2024b) to model resume-job compatibility. Similar to ConFit, we use an encoder to produce an embedding of a given resume or job post, and model the matching score between an $\langle R, J \rangle$ pair as the cosine similarity of their representations. Unlike ConFit, ConFit v2 uses a simpler encoder architecture, and substantially improves ranking performance by using 1) hypothetical reference resume generated by an LLM to help improve the encoder's job data understanding; and 2) a novel hard-negative mining method to enhance encoder contrastive learning. We provide a high-level overview of ConFit v2 during inference in Figure 2. We detail each modification below.

3.1 Encoder Architecture

Many prior works on resume-job matching employ complex neural architectures to encode human-designed matching heuristics (Zhu et al., 2018; Yang et al., 2022). For example, Shao et al. (2023); Yu et al. (2024b) uses a hierarchical attention mechanism to model interactions between each *section* (e.g., Education, Experience, etc.) of a resume and a job post. However, in practice, these modifications often rely on specific domain knowledge and require additional structural constraints on resume/job data, restricting their generalizability.

Different from these methods, we treat the resume/job post as *a single sequence of text*, and directly use a transformer-based encoder to produce the embedding of the entire document. This allows the encoder model itself to learn the most relevant features from the data, instead of relying on human-designed heuristics. We illustrate this difference in Figure A1. In practice, we find this simpler design is more effective and robust across different datasets and backbones (Section 4.6). We denote this simplified encoder as ConFit*.

3.2 Hypothetical Resume Embedding

In traditional dense passage retrieval tasks, one challenge is the discrepancy between query and passage formats and content (Gao et al., 2022;

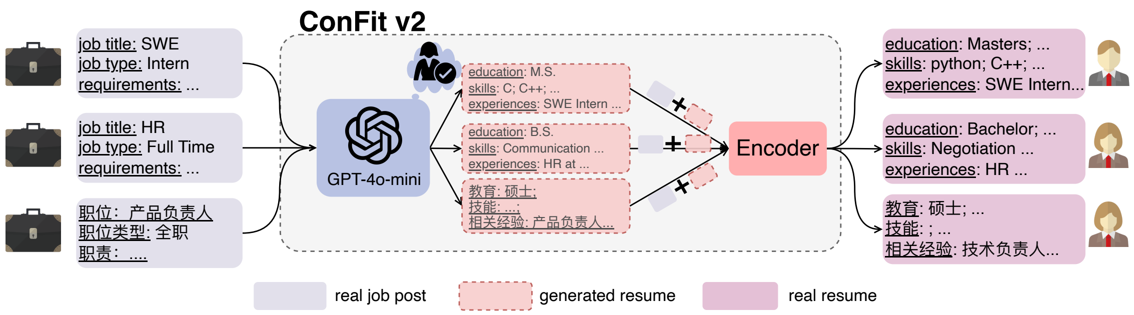

Figure 2: CONFIT V2 inference. Given a job post, we first use an LLM to generate a hypothetical reference resume given the job post, and then outputs a job embedding using the concatenation of the generated resume and the job post. Given a resume, CONFIT V2 outputs a resume embedding directly using our trained encoder model. Finally, cosine similarity is used to compute the compatibility between the input resume and job post.

Zhou et al., 2024). To address this, Gao et al. (2022) shows that zero-shot passage ranking improves significantly with Hypothetical Document Embeddings (HyDe), where an LLM generates a hypothetical passage from a query, and the encoder ranks actual passages based on their cosine similarity to the hypothetical one. We observe a similar discrepancy in resume-job matching, and find a substantial performance gain when high-quality (real) resumes are used in place of jobs during ranking. See Appendix B for more details.

Given a dataset of accepted/rejected resume-job pairs, one extension to systems like HyDe is to fine-tune an LLM such as LLaMA-3 (Grattafiori et al., 2024) or Qwen-2.5 (Team, 2024) using SFT and DPO (Rafailov et al., 2024). However, in our prior study, we find fine-tuning these models yields inferior results (see Appendix B.4 for more details and empirical results) to few-shot prompting powerful closed-source models such as GPT-4o-mini (OpenAI, 2024a). We believe this is because accepting a resume can be subjective, making it hard for an LLM to learn an "ideal candidate" from the dataset. We thus construct hypothetical references by *augmenting the existing job posts*. Specifically, we 1) fix N accepted resume-job pairs from the training set as few-shot examples; 2) prompt GPT-4o-mini to generate a reference resume given a job post; and 3) concatenate the generated resume with the real job post. We repeat this for all jobs used during training and testing.

3.3 RUNNER-UP MINING

To accurately model resume-job compatibility, the encoder needs to distinguish between matching and near-matching resume-job pairs. CONFIT (Yu et al., 2024b) achieves state-of-the-art performance by training the encoder with contrastive loss, where given a job the model uses 1) random resumes as in-batch negatives and 2) rejected resumes as hard negatives, and vice versa for resumes. However, due to label scarcity (Section 2), the quantity of rejected resumes/hard negatives is highly limited.

We propose RUNNER-UP MINING (RUM), a new hard-negative mining method that selects "runner-up" resume-job pairs as hard negatives based on compatibility scores from an encoder model. In dense text retrieval tasks, many prior hard-negative mining methods that treat *unlabeled or incorrect top-k results* (e.g., using BM25) as hard *negatives* (Xiong et al., 2021; Zhao et al., 2024). However, in resume-job matching, we find many of these top-k results are *positive* pairs that are unlabeled simply because the candidate did not apply to the job.[1] To avoid these top-k false negatives, we instead take "runner-up" (e.g., top 3%-4%) samples based on the cosine similarity scores, and use them as hard negatives for training.

Specifically, RUM begins by computing the cosine similarity scores between all possible resume-job pairs using an encoder model f_μ (e.g., CONFIT*). Then, we rank these pairs by similarity score and randomly sample resumes and jobs from the top percentile *ranges* (e.g., top 3%-4%) as challenging hard negatives. Finally, during contrastive training, we replace hard negatives used in CONFIT (i.e., rejected resumes/jobs) with these hard negatives mined by RUM.

[1] For example, he/she may be overqualified or may have already accepted another offer.

3.4 ConFit v2

We summarize our contribution in ConFit v2. First, we simplified the encoder architecture used by ConFit, so that complex dynamics between different resumes and job posts are learned directly from the data (i.e., ConFit*, Section 3.1). Then, we improve the encoder training process by using 1) HyRe to generate pivot hypothetical resumes to simplify the job post representation space during both training and testing; and 2) RUM to mine high-quality hard negatives for more effective contrastive training. We illustrate ConFit v2's inference process in Figure 2. Given a job post, ConFit v2 converts it into an embedding by first using an LLM to generate a hypothetical resume, and then using the encoder to produce a job embedding using the concatenation of the generated resume and the original job post. Given a resume, ConFit v2 directly uses the encoder to produce the resume embedding. The compatibility score between a resume and a job is then computed as the cosine similarity between their embeddings.

To train ConFit v2 encoder, we use HyRe to augment all job posts with generated reference resumes (Section 3.2), and replace them with original job posts for later training and testing. Then, we use RUM to mine hard-negative resumes and (augmented) jobs (Section 3.3). Finally, we train the simplified ConFit* using the modified contrastive learning loss $\mathcal{L}$ from Yu et al. (2024b):

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_J \tag{1}$$

$$\mathcal{L}_R = -\log \frac{e^{f_\theta(R_i^+, J_i^+)}}{e^{f_\theta(R_i^+, J_i^+)} + \sum_{j=1}^{l} e^{f_\theta(R_i^+, J_{i,j}^-)}}$$

$$\mathcal{L}_J = -\log \frac{e^{f_\theta(R_i^+, J_i^+)}}{e^{f_\theta(R_i^+, J_i^+)} + \sum_{j=1}^{l} e^{f_\theta(R_{i,j}^-, J_i^+)}}$$

where $\mathcal{L}_R$ and $\mathcal{L}_J$ is the resume/job contrastive loss, respectively; (R^+, J^+) denotes accepted resume-job pairs; (R^-, J^+) or (R^+, J^-) denote negative resume-job pairs including both in-batch negatives and hard negatives; and f_θ is cosine similarity after embedding the resume-job pairs.

4 Experiments

We evaluate ConFit v2 on two real-world person-job fit datasets, and measure its performance on ranking resumes and ranking jobs.

4.1 Dataset and Preprocessing

AliYun Dataset To our knowledge, the 2019 Alibaba job-resume intelligent matching competition[2] *provided* the only publicly available person-job fit dataset. All resume and job posts were desensitized and were processed into a collection of text fields, such as "Education" and "Work Experiences" for a resume (see Appendix C for more details). All resumes and jobs are in Chinese.

Recruiting Dataset The resumes and job posts are provided by a hiring solution company. To protect the privacy of candidates, all records have been anonymized by removing sensitive identity information (see Appendix C). For each resume-job pair, we record whether the candidate is accepted ($y = 1$) or rejected ($y = 0$) for an interview. Similar to the AliYun dataset, all resumes and jobs were available as a collection of sections/fields. Both English and Chinese resumes and jobs are included.

4.2 Implementation Details

We follow ConFit (Yu et al., 2024b) and experiment with two encoder architectures. We consider E5-base (Wang et al., 2022), and Jina-v2-base. (Mohr et al., 2024). Both encoders are trained on large-scale Chinese-English bilingual text data, and were amongst the best embedding models according to benchmarks such as MTEB (Muennighoff et al., 2022) at the time of the project. For HyRe, we use GPT-4o-mini as the LLM for generating hypothetical resumes due to its cost-effectiveness. We provide the prompts used in Appendix B.1. We then train ConFit* with HyRe using contrastive learning (without hard negatives), and use to mine hard negatives with RUM. For RUM, we use the top 3%-4% percentile to pick hard-negatvies *across all experiments*. Finally, we train ConFit* again, using 1) HyRe augmented job data; and 2) RUM mined hard negatives. We use Eq (1) with a batch size of 4 and 2 hard negatives per batch, and a learning rate of 1e-5. This is the final encoder used for ConFit v2. We run all experiments using one A100 80GB GPU. To speed up cosine similarity computation across a large pool of resumes and jobs, we use FAISS (Johnson et al., 2019) throughout all experiments.

4.3 Baselines

We compare ConFit v2 against both recent best person-job fit systems and strong baselines from

[2] https://tianchi.aliyun.com/competition/entrance/231728

Method	Encoder	Recruiting Dataset				AliYun Dataset			
		Rank Resume		Rank Job		Rank Resume		Rank Job	
		Recall@100	nDCG@100	Recall@10	nDCG@10	Recall@10	nDCG@10	Recall@10	nDCG@10
RawEmbed.	E5-base	46.48	22.87	43.58	25.63	49.52	29.66	36.24	27.19
	text-embedding-3-large	75.49	40.28	79.67	56.73	68.74	46.19	48.52	38.36
BM25	-	41.73	17.36	38.92	24.14	63.18	40.56	44.83	31.18
MV-CoN	E5-base	10.29	7.46	24.75	8.29	11.01	6.47	14.63	8.29
InEXIT	E5-base	12.32	6.09	19.33	4.16	10.54	5.15	9.82	6.11
CONFIT	E5-base	65.13	28.74	68.42	40.85	88.08	70.39	61.58	43.45
CONFIT V2 (ours)	E5-base	**84.44**	**48.40**	**88.67**	**66.61**	**96.18**	**82.20**	**82.11**	**67.96**

Table 1: Comparing ranking performance of various approaches with E5-base as backbone encoder. Results for non-deterministic methods are averaged over 3 runs. Best result is shown in **bold**.

Method	Encoder	Recruiting Dataset				AliYun Dataset			
		Rank Resume		Rank Job		Rank Resume		Rank Job	
		Recall@100	nDCG@100	Recall@10	nDCG@10	Recall@10	nDCG@10	Recall@10	nDCG@10
RawEmbed.	Jina-v2-base	54.73	26.34	44.38	27.64	57.20	36.72	43.27	32.29
	text-embedding-3-large	75.49	40.28	79.67	56.73	68.74	46.19	48.52	38.36
BM25	-	41.73	17.36	38.92	24.14	63.18	40.56	44.83	31.18
MV-CoN	Jina-v2-base	10.50	4.06	19.83	5.72	9.11	4.72	13.50	9.99
InEXIT	Jina-v2-base	8.83	4.01	6.50	3.05	11.36	4.83	13.49	10.46
CONFIT	Jina-v2-base	71.28	34.79	76.50	52.57	87.81	65.06	72.39	56.12
CONFIT V2 (ours)	Jina-v2-base	**86.13**	**51.90**	**94.25**	**73.32**	**97.07**	**83.11**	**80.49**	**68.02**

Table 2: Comparing ranking performance of various approaches with Jina-v2-base as backbone encoder. Results for non-deterministic methods are averaged over 3 runs. Best result is shown in **bold**.

information retrieval systems.

Prior person-job fit systems include *MV-CoN* (Bian et al., 2020), *InEXIT* (Shao et al., 2023), and *ConFit* (Yu et al., 2024b) and more (Qin et al., 2018; Zhu et al., 2018; Yang et al., 2022). MV-CoN considers a co-teaching network (Han et al., 2018) to learn from sparse, noisy person-job fit data, and InEXIT uses hierarchical attention to model interactions between the text fields of a resume-job pair. CONFIT focuses on dense retrieval techniques, and uses contrastive learning and LLM-based data augmentation to train an encoder. Other older methods such as APJFNN (Qin et al., 2018), PJFNN (Zhu et al., 2018), and DPGNN (Yang et al., 2022) are omitted since they are already outperformed by these more recent methods.

We also compare against methods from information retrieval systems such as: *BM25* (Robertson and Zaragoza, 2009; Trotman et al., 2014) and *RawEmbed*. BM25 is a strong baseline used for many text ranking tasks (Thakur et al., 2021; Kamalloo et al., 2023), and RawEmbed is based on dense retrieval methods (Karpukhin et al., 2020; Johnson et al., 2019) that directly uses a pre-trained encoder to produce dense embeddings for inner product scoring. Since RawEmbed does not need training, we consider both open-source encoders such as E5 (Wang et al., 2022) and Jina-v2 (Mohr et al., 2024) and commercial models such as OpenAI's text-embedding-003-large (OpenAI, 2024b).

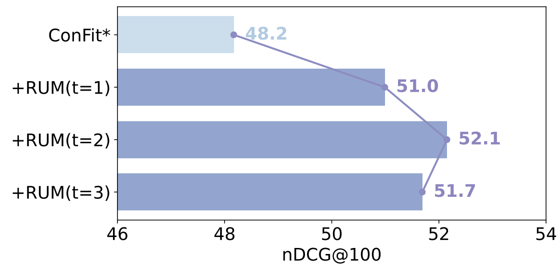

Figure 3: Iterative RUM using Jina-v2-base as encoder model. "RUM(t=N)" indicates applying RUM N times.

4.4 Main Results

Table 1 summarizes CONFIT V2's performance in comparison to other baselines on the Recruiting and AliYun datasets when E5-base is used as the encoder. We report two ranking metrics, Recall@K and nDCG@K. In general, CONFIT shows substantial improvement over many prior methods, such as MV-CoN, InEXIT, and BM25. CONFIT V2 further improves CONFIT by 17.1% in recall and 20.4% in nDCG score on average, outperforming all other baselines, including OpenAI's text-embedding-003-large, on both datasets.

Table 2 summarizes each method's performance when a different encoder (Jina-v2-base) is used. Similar to Table 1, CONFIT shows impressive performance compared to many baselines, but is exceeded by OpenAI's text-embedding-003-large. However, CONFIT V2 outperforms all other base-

Method	Recruiting Dataset				AliYun Dataset			
	Rank Resume		Rank Job		Rank Resume		Rank Job	
	Recall@100	nDCG@100	Recall@10	nDCG@10	Recall@100	nDCG@100	Recall@10	nDCG@10
ConFit	71.28	34.79	76.50	52.57	87.81	65.06	72.39	56.12
ConFit*	82.53	48.17	85.58	64.91	94.90	78.40	78.70	65.45
+HyRe	85.28	49.50	90.25	70.22	96.62	81.99	**81.16**	67.63
+RUM	**86.13**	**51.90**	**94.25**	**73.32**	**97.07**	**83.11**	80.49	**68.02**

Table 3: ConFit v2 ablation studies. "ConFit*" refers using a simplified encoder architecture. ConFit v2 trains ConFit* with RUM and HyRe. We use Jina-v2-base as the encoder due to its better performance.

lines, by up to 10.6% in recall and 14.5% in nDCG score for ranking resumes and jobs on both datasets[3]. We believe these results underscore the effectiveness of our method across different encoders and datasets.

4.5 Iterative RUM

Since RUM only requires an encoder model to mine hard negatives, we also experiment with using RUM iteratively to improve the model's performance. For each iteration, we 1) use the encoder trained from the previous iteration to mine hard negatives using RUM; and 2) train the backbone encoder again using the newly acquired hard negatives.[4] Throughout all iterations, we keep all hyperparameters the same as in Section 4.2. We focus on the ranking resume task in the Recruiting dataset since it is the most challenging.

We present the results in Figure 3. In general, we find that training with more than one iteration ($t > 1$) improves resume ranking performance compared to a single iteration ($t = 1$). However, as the number of iterations increases, improvement begins to drop. We believe this is similar to many model self-improvement research (Huang et al., 2023; Yu et al., 2024a). While model-created data often helps improve performance, these data also contains noises that can compound over multiple iterations. We believe techniques such as model ensembles (Zhang et al., 2011; de Souza P. Moreira et al., 2024) could help mitigate this issue, and we leave this for future work.

4.6 Ablation Studies

Table 3 presents our ablation studies for each component of ConFit v2. First, we consider only simplifying the encoder architecture (denoted as *ConFit**), and use rejected resumes/jobs as hard negatives used by ConFit (Yu et al., 2024b). We find that this simplification significantly improves ranking performance. We believe this is due to the strong performance of recent embedding models, which can effectively learn complex interactions between resumes and job posts directly from data. Next, we use HyRe to generate reference resumes, augment job posts, and train ConFit* on the augmented data (denoted as +HyRe). We find this to improve 2.9% absolute in recall and 3.1% nDCG scores on average. Finally, we use RUM to mine hard resume/(augmented) job negatives (denoted as *+RUM*), and replace them with the rejected resume/jobs used by ConFit. In Table 3, we find that RUM further improved by, on average, 1.2% absolute in recall scores, and 1.8% absolute in nDCG across both datasets. We find that all components are important for ConFit v2.

For other results, such as comparing RUM against methods that use BM25, with different percentile ranges other than the top 3%-4% as well as testing RUM against hard negatives mined by BM25, please refer to Section 5.2.

5 Analysis

In this section, we provide both qualitative and quantitative analysis on the ranking output produced by ConFit v2. We mainly focus on the Recruiting dataset as it is more challenging.

5.1 Error Analysis

To analyze the errors made by ConFit v2, we manually inspect 30 *negative* resume-job pairs from the ranking tasks that are *incorrectly ranked at top 5%* and are before at least one positive pair. For each incorrectly ranked pair, we compare against other positive resume-job pairs from the dataset, and categorize the errors following the criteria from Yu et al. (2024b): *unsuitable*, where some requirements in the job post are not satisfied by the re-

[3]We believe this is due to, despite Jina-v2's recency, Günther et al. (2023) find E5 to be more robust for retrieval tasks.

[4]We also experimented with continuously training the model from the previous iteration, but found that it can more easily lead to overfitting.

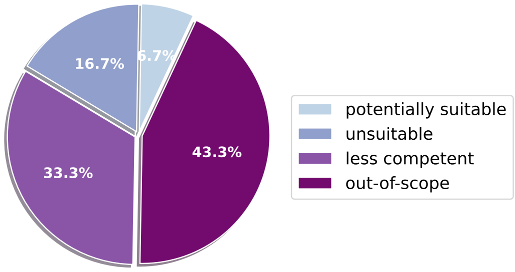

Figure 4: ConFit v2 error analysis. We find 43% of the errors made are due to reasons not identifiable using resume/job documents alone, and 33% due to a candidate's resume satisfying all the job requirements but is less competent than other competing candidates

	Recruiting Dataset			
	Rank Resume		Rank Job	
Method	Recall@100	nDCG@100	Recall@10	nDCG@10
BM25(top-10)	82.95	44.62	86.75	64.36
STAR	82.70	47.19	88.67	68.55
RUM(0%-1%)	80.80	48.41	86.00	66.17
RUM(1%-2%)	85.13	49.92	85.13	66.85
RUM(2%-3%)	83.08	49.09	90.25	70.21
RUM(3%-4%)	85.43	**50.99**	**91.38**	**71.34**
RUM(4%-5%)	**85.56**	50.88	87.25	69.28

Table 4: Comparing RUM with BM25 methods, as well as RUM under different percentiles ranges. We denote this as "RUM(*L%-H%*)". We implement the "STAR" method following Zhan et al. (2021).

Method	**Gender**	**Male(%)**	**Female(%)**
ConFit*	✗	70.43	29.57
ConFit*	✓	75.63	24.37
ConFit* + RUM	✗	66.67	33.33
ConFit* + RUM	✓	69.72	30.28

Table 5: Gender distribution in the top-10 ranked resume, averaged across all test job postings. "Gender" indicates whether gender information is used during training. "Male/Female" indicates the proportion of the top-10 resume that belongs to a male/female candidate.

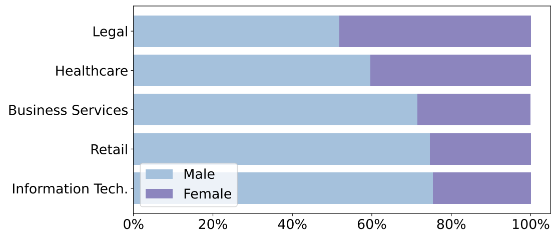

Figure 5: Gender distribution in different industries. We picked five industries to illustrate trend. In entire dataset, 72% of the candidates are male, and 28% are female.

sume; *less competent*, where a resume satisfies all job requirements, but many competing candidates have a higher degree/more experience; *out-of-scope*, where a resume satisfies all requirements, appears competitive compared to other candidates, but is still rejected due to other (e.g., subjective) reasons not presented in our resume/job data themselves; and *potentially suitable*, where a resume from the ranking tasks satisfied the requirements and seemed competent, but had no label in the original dataset.

We present our analysis in Figure 4. We find a significant portion of the errors are *out-of-scope*, where it is hard to determine the reason for rejection based on information in the resume/job data alone. The next most frequent error is *less competent*, which is understandable since ConFit v2 produces a compatibility score for a resume-job pair independent of other candidates. Lastly, we also find that 16.7% of the errors are *unsuitable*, with resumes not fulfilling certain requirements such as "Backend SWEs with *Ruby* experience". We believe *unsuitable* errors can be mitigated by combining ConFit v2 with keyword-based approaches (e.g., BM25 or human-designed rules). We leave this for future work.

5.2 Additional Results on RUM

Since RUM uses hyperparameters such as percentile ranges, we additionally investigate: (1) how different percentile ranges affect performance, and (2) how RUM compares to other BM25-based methods (Karpukhin et al., 2020; Zhao et al., 2024) or neural-based methods (Zhan et al., 2021; Xiong et al., 2021) for hard-negative mining. Specifically, we consider "BM25(top-10)", which retrieves top-10 highest-scoring unlabeled or incorrect resume/job computed by BM25; and STAR from Zhan et al. (2021) which uses top resume/jobs found by a retriever model.

We present the result in Table 4. Overall, we find RUM consistently surpasses BM25-based negatives in nDCG across both ranking tasks. When using RUM, we find although ranges such as 0%–1% underperforms BM25 in some metrics, using ranges such as 2%-3%, and 3%-4% significantly underperforms BM25. This indicates that many high-scoring unlabeled samples are likely *positives* for person-job fit. In this work, we picked 3%-4% due to its high average performance, and *fixed it for all subsequent runs* (e.g., with different model architectures and datasets).

5.3 Bias Analysis

ConFit v2 relies on pretrained encoders such as E5 and Jina-v2 (Devlin et al., 2019; Wang et al., 2022), and it is well-known that many powerful encoders can contain biases (Brunet et al., 2019; May et al., 2019; Jentzsch and Turan, 2022; Caliskan et al., 2022) such as gender, age, demographics, etc. As a case study, we examine whether ranking outputs from ConFit v2 contain gender bias. For this experiment, we add gender information[5] back to our desensitized resumes, and compare 1) whether training on gender-included data increases gender disparities in the ranking output; and 2) whether RUM has any effect on gender bias.

We present our analysis in Table 5. Across all experiments, we find adding gender information back into the training data increases gender disparities in the ranking output. "ConFit*" trained with gender information widened the gender gap in the ranking output by ~10% compared to training without gender, and "ConFit*+RUM" widened the gap by ~6%. In general, we believe one major cause of this disparity is the inherent biases in the datasets. In Figure 5, we find gender distributions are uneven across different industries, and that overall, 72% of the (accepted or rejected) resumes used during training are from male candidates, and only 28% are from female candidates.

In Table 5, we also find that using RUM helps reduce gender disparities compared to ConFit* alone. We believe this indicates RUM provided challenging hard negatives that do *not* always rely on gender information, hence reducing gender disparity during training. In general, we advise researchers and practitioners to be cautious when using ConFit v2, and to use explicit de-biasing techniques (Bolukbasi et al., 2016; Cheng et al., 2021; Gaci et al., 2022; Guo et al., 2022) *in addition to* data de-sensitization used in this work. We do not condone the use of ConFit v2 for any ethically unjust purposes.

6 Related Work

Person-job fit systems Early person-job fit systems that use neural networks typically focus on architecture designs. These methods include Qin et al. (2018); Zhu et al. (2018); Rezaeipourfarsangi and Milios (2023); Jiang et al. (2020); Mhatre et al. (2023), which explores architectures such as RNN, LSTM (Staudemeyer and Morris, 2019) and CNN (O'Shea and Nash, 2015). However, these are significantly outperformed by more recent transformer-based methods such as Maheshwary and Misra (2018); Rezaeipourfarsangi and Milios (2023); Bian et al. (2019); Zhang et al. (2023), which focuses on small architecture or loss modifications. For example, MV-CoN (Bian et al., 2020) uses a co-teaching network (Malach and Shalev-Shwartz, 2018) to perform gradient updates based on model's confidence to data noises; In-EXIT (Shao et al., 2023), which uses hierarchical attention to model resume-job interactions; and DPGNN (Yang et al., 2022), which uses a graph-based approach with a novel BPR loss to optimize for resume/job ranking. However, these methods tend to focus on task-specific modifications, such as assuming no unseen resumes/jobs at test time or assuming access to internal data such as whether the recruiter sent private messages to the job applicant. More recently, ConFit (Yu et al., 2024b) focuses on applying dense retrieval techniques to person-job fit. Using contrastive learning with encoders such as E5 (Wang et al., 2022), Yu et al. (2024b) demonstrates its performance and flexibility by achieving the best scores in almost all ranking and classification tasks across two different person-job fit datasets. In this work, we propose HyRe and RUM to further enhance ConFit v2, achieving an average absolute improvement of 13.8% in recall and 17.5% in nDCG across job ranking and resume-ranking tasks in two resume-job benchmarks.

Dense retrieval techniques ConFit v2 benefits from dense retrieval techniques such as contrastive learning (Chen et al., 2020; Radford et al., 2021) and hard-negative mining. Popular methods in text retrieval include BM25 (Robertson and Zaragoza, 2009; Trotman et al., 2014), a keyword-based approach used as the baseline in many text ranking tasks (Nguyen et al., 2016; Thakur et al., 2021; Muennighoff et al., 2022), and dense retrieval methods such as Karpukhin et al. (2020); Izacard et al. (2021); Wang et al. (2022); Günther et al. (2023), which uses contrastive learning with an encoder to obtain high-quality passage embeddings, and typically performs top-k search based on inner product.

To further improve retrieval results, researchers considered methods such as query/document expansions (Carpineto and Romano, 2012) and hard-negative mining (Robinson et al., 2021). Recent

[5]Since gender information is voluntarily submitted by the candidates, this affects around 15% of all resumes.

query/document expansion approaches include HyDE (Gao et al., 2022) and HyQE (Zhou et al., 2024), which prompts an LLM to augment the original query or the passage to retrieve and is typically used *without* finetuning (the LLM or the encoder model). Common hard-negative mining strategies often use BM25 to select top-k unlabeled/incorrect samples as hard negatives (Karpukhin et al., 2020; Zhao et al., 2024; Nguyen et al., 2023). Other methods include: ANCE (Xiong et al., 2021) which samples negatives from the top retrieved documents asynchronously during training; SimANS (Zhou et al., 2022) which samples hard negatives using a manually designed probability distribution; and more (Zhan et al., 2021; Li et al., 2024). CONFIT (Yu et al., 2024b) presents the first successful attempt to use contrastive learning for person-job fit. We extend CONFIT and adapt recent dense retrieval techniques to person-job fit by 1) sampling hard-negative resumes/jobs from a "runner-up" *percentile ranges* to mitigate selecting false-negatives; and 2) few-shot prompting an LLM to augment a job posted used for later encoder *training*.

7 Conclusion

We propose CONFIT V2, an improvement of CONFIT to model person-job fit. Similar to CONFIT, we model person-job fit using dense retrieval techniques such as contrastive learning. Unlike CONFIT, we first simplified the encoder architecture, and improved encoder training using 1) hypothetical reference resume generated by an LLM to augment a job post; and 2) a new hard-negative mining strategy that selects "runner-up" resume-job pairs to avoid false negatives used for encoder training. We evaluate CONFIT V2 on two real-world datasets, and demonstrate that it outperforms CONFIT and prior methods, achieving an average *absolute* improvement of 13.8% in recall and 17.5% in nDCG across both ranking resume and ranking job tasks. We believe our work lays a strong foundation for future resume-job matching systems to leverage the latest advancements in dense text retrieval.

8 Limitations

Data Sensitivity To our knowledge, there is no standardized, public person-job fit dataset.[6] that can be used to compare performances of existing systems (Zhu et al., 2018; Qin et al., 2018; Bian et al., 2020; Yang et al., 2022; Shao et al., 2023) This is due to the highly sensitive nature of resume and job post content, making large-scale person-job fit datasets largely proprietary. We follow Yu et al. (2024b) and provide the best effort to make CONFIT V2 reproducible and extensible for future work. We will open-source implementations of CONFIT V2, related baselines, data processing scripts, and dummy train/valid/test data files that can be used to test drive our system end-to-end. We will also privately release full datasets to researchers under appropriate license agreements, aiming to make future research in person-job fit more accessible.

Local HYRE During HYRE, we use a commercial LLM such as GPT-4o-mini to generate hypothetical reference resumes to augment our job data for retrieval. Ideally, one would prefer using an in-house, local model to better retrain data privacy and optimize for inference speed. However, in our prior study, we find fine-tuning open-source LLMs (up to 32B in size) for this task to be challenging, due to the scarce and subjective nature of the person-job fit labels (Appendix B.4). We believe designing data/training algorithms tailored for this task could benefit person-job fit systems such as CONFIT V2, which we leave for future work.

Other Hard-Negative Mining Methods In this work, we proposed RUM to select *near* top resume/jobs as hard-negatives, in order to prevent selecting potentially positive unlabeled pairs for training. We compared RUM against other relevant hard-negative mining methods from traditional information retrieval domains such as BM25-based (Karpukhin et al., 2020; Zhao et al., 2024) and STAR (Zhan et al., 2021), and found RUM outperforms them in the person-job fit domain. However, as research in retrieval progresses, we believe there will be more novel information retrieval methods that could enhance modeling person-job fit. As we continue to improve CONFIT V2, we plan to experiment with and adapt these new SOTA methods. We leave this for future work.

9 Ethical Considerations

CONFIT V2 uses and trains pretrained encoders such as E5 (Wang et al., 2022) and Jina-v2 (Günther et al., 2023), and it is well-known that these models contain biases including but not limited to gender, race, demographics, and more (Brunet et al., 2019; May et al., 2019; Jentzsch and Turan,

[6]The AliYun dataset used in this work is no longer publicly available as of 09-11-2023.

2022; Caliskan et al., 2022). For practical person-job fit systems, we believe it is crucial to ensure that they do **not** exhibit these biases, such as preferring a certain gender for certain jobs. In this work, we have followed best practices from prior work (Yang et al., 2022; Yu et al., 2024b) to remove sensitive information (e.g., gender) from our data. However, our bias analysis (Section 5.3) reveals that gender disparities in the trained model could stem from imbalances in the dataset. We do **not** condone using CONFIT V2 for real-world applications without using de-biasing techniques such as Bolukbasi et al. (2016); Cheng et al. (2021); Gaci et al. (2022); Guo et al. (2022); Schick et al. (2021), and in general, we do **not** condone the use of CONFIT V2 for any morally unjust purposes. To our knowledge, there is little work on investigating or mitigating biases in existing person-job fit systems, and we believe this is a crucial direction for future person-job research.

References

Shuqing Bian, Xu Chen, Wayne Xin Zhao, Kun Zhou, Yupeng Hou, Yang Song, Tao Zhang, and Ji-Rong Wen. 2020. Learning to match jobs with resumes from sparse interaction data using multi-view co-teaching network. *Preprint*, arXiv:2009.13299.

Shuqing Bian, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. 2019. 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)*, pages 4810–4820, Hong Kong, China. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Preprint*, arXiv:1607.06520.

Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard Zemel. 2019. Understanding the origins of bias in word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 803–811. PMLR.

Aylin Caliskan, Pimparkar Parth Ajay, Tessa Charlesworth, Robert Wolfe, and Mahzarin R. Banaji. 2022. Gender bias in word embeddings: A comprehensive analysis of frequency, syntax, and semantics. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22. ACM.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1).

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *Preprint*, arXiv:2002.05709.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders. *Preprint*, arXiv:2103.06413.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *Preprint*, arXiv:2407.15831.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Yacine Gaci, Boualem Benatallah, Fabio Casati, and Khalid Benabdeslem. 2022. Debiasing pretrained text encoders by paying attention to paying attention. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9582–9602, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *Preprint*, arXiv:2212.10496.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023, Dublin, Ireland. Association for Computational Linguistics.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *Preprint*, arXiv:2310.19923.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Preprint*, arXiv:1804.06872.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Mansoor Iqbal. 2025. LinkedIn usage and revenue statistics (2023). online. Accessed: 2025-02-02.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Sophie Jentzsch and Cigdem Turan. 2022. Gender bias in BERT - measuring and analysing biases through sentiment rating in a realistic downstream classification task. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 184–199, Seattle, Washington. Association for Computational Linguistics.

Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning effective representations for person-job fit by feature fusion. *Preprint*, arXiv:2006.07017.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2023. Resources for brewing beir: Reproducible reference models and an official leaderboard. *Preprint*, arXiv:2306.07471.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Xiaopeng Li, Xiangyang Li, Hao Zhang, Zhaocheng Du, Pengyue Jia, Yichao Wang, Xiangyu Zhao, Huifeng Guo, and Ruiming Tang. 2024. Syneg: Llm-driven synthetic hard-negatives for dense retrieval. *Preprint*, arXiv:2412.17250.

Saket Maheshwary and Hemant Misra. 2018. Matching resumes to jobs via deep siamese network. *Companion Proceedings of the The Web Conference 2018*.

Eran Malach and Shai Shalev-Shwartz. 2018. Decoupling "when to update" from "how to update". *Preprint*, arXiv:1706.02613.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Sonali Mhatre, Bhawana Dakhare, Vaibhav Ankolekar, Neha Chogale, Rutuja Navghane, and Pooja Gotarne. 2023. Resume screening and ranking using convolutional neural network. In *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pages 412–419.

Isabelle Mohr, Markus Krimmel, Saba Sturua, Mohammad Kalim Akram, Andreas Koukounas, Michael Günther, Georgios Mastrapas, Vinit Ravishankar, Joan Fontanals Martínez, Feng Wang, et al. 2024. Multi-task contrastive learning for 8192-token bilingual text embeddings. *arXiv preprint arXiv:2402.17016*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Thanh-Do Nguyen, Chi Minh Bui, Thi-Hai-Yen Vuong, and Xuan-Hieu Phan. 2023. Passage-based BM25 hard negatives: A simple and effective negative sampling strategy for dense retrieval. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 591–599, Hong Kong, China. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

OpenAI. 2024a. GPT-4o-mini: Advancing cost-effective intelligence.

OpenAI. 2024b. New embedding models and api updates.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *Preprint*, arXiv:1511.08458.

Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. 2018. 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*, SIGIR '18, page 25–34, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Sima Rezaeipourfarsangi and Evangelos E. Milios. 2023. Ai-powered resume-job matching: A document ranking approach using deep neural networks. In *Proceedings of the ACM Symposium on Document Engineering 2023*, DocEng '23, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Taihua Shao, Chengyu Song, Jianming Zheng, Fei Cai, and Honghui Chen. 2023. Exploring internal and external interactions for semi-structured multivariate attributes in job-resume matching. In *International Journal of Intelligent Systems*.

Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm – a tutorial into long short-term memory recurrent neural networks. *Preprint*, arXiv:1909.09586.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

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*.

Xiao Yu, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhou Yu. 2024a. Teaching language models to self-improve through interactive demonstrations. *Preprint*, arXiv:2310.13522.

Xiao Yu, Jinzhong Zhang, and Zhou Yu. 2024b. Confit: Improving resume-job matching using data augmentation and contrastive learning. RecSys '24, page 601–611, New York, NY, USA. Association for Computing Machinery.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA. Association for Computing Machinery.

Peng Zhang, Xingquan Zhu, Yong Shi, Li Guo, and Xindong Wu. 2011. Robust ensemble learning for mining noisy data streams. *Decision Support Systems*, 50(2):469–479.

Yunchong Zhang, Baisong Liu, and Jiangbo Qian. 2023. Fedpjf: federated contrastive learning for privacy-preserving person-job fit. *Applied Intelligence*, 53:27060 – 27071.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4).

Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-rong Wen, and Nan Duan. 2022. SimANS: Simple ambiguous negatives sampling for dense text retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 548–559, Abu Dhabi, UAE. Association for Computational Linguistics.

Weichao Zhou, Jiaxin Zhang, Hilaf Hasson, Anu Singh, and Wenchao Li. 2024. Hyqe: Ranking contexts with hypothetical query embeddings. *Preprint*, arXiv:2410.15262.

Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *Preprint*, arXiv:1810.04040.

A More Details on Model Architecture

Figure A1 displays encoder model architecture used in ConFit and ConFit*. We remove all the additional layers used after the backbone encoder, including the attention layers that interact among features of each field, and the MLP layers for feature fusion. Instead, we directly employ mean pooling on the embeddings of all the tokens in the texts to obtain document-level representation (Wang et al., 2022; Günther et al., 2023). Then, we perform contrastive learning on the document-level embeddings.

We implement this with both encoders such as E5 (Wang et al., 2022) and long-context encoders such as Jina-v2 (Günther et al., 2023). E5 supports 512 input sequence length and Jina-v2 supports 8192. We add L_2 normalization to the features, and use temperature T to rescale the cosine similarity to facilitate training. For E5 we use $T = 0.01$, and for Jina-v2 we use $T = 0.05$. In Table 1 and Table 2, we find that ConFit* significantly improve upon ConFit across all settings.

B Additional Results on HyRe

B.1 HyRe Prompts

We use the prompt in Table A7 to perform few-shot prompting for hypothetical resume generation. In the prompt, *{{{example job}}}*, *{{{example resume}}}* are randomly sampled positive pairs from the training set, and the *{{{target job}}}* is the target job to be augmented. For the ablation study in B.3, we swap the order of template job and resume, send in a resume as target, and revise the prompt correspondingly. When concatenating job and hypothetical resume, we add *["An Example Resume"]* in between for clarity.

B.2 HyRe Upperbound

In this section, we estimate the upper bound performance of HyRe by using real resumes from the dataset to augment each job during both training and testing. However, evaluation based on training the entire dataset is time-consuming, and thus we trained on a randomly sampled subset containing 20% of the training data while keeping all other settings identical to those used for training on the full dataset.

We experiment with three strategies of selecting "ideal" reference resumes for each job from the dataset. For *max-match (human)*, we select the highest-scoring resume for each job according to additional annotations provided by a hiring solution company. For *max-match (Jina)*, we select the highest-scoring resume that achieves high human annotation score **and** a high cosine similarity score computed directly by Jina-v2 (Günther et al., 2023). For *centroid*, we select the centroid resume of all accepted resumes given a job post. Since computing the centroid resume across the full resume pool is time-consuming, we instead restrict the pool to all *labeled* resumes given the job post. Then, we pick the resume, which has the shortest average L_2 normalized distance (i.e. cosine similarity) to its job's accepted resumes, as the centroid resume.

	Recruiting Dataset			
	Rank Resume		Rank Job	
Method	Recall@100	nDCG@100	Recall@10	nDCG@10
ConFit*	79.00	44.21	80.83	60.67
Max-match (human)	82.06	73.14	84.75	70.11
Max-match (Jina)	80.14	65.95	82.78	66.34
Centroid	74.82	68.28	87.50	69.29

Table A1: Estimating HyRe upperbound using real resumes. To speed-up evaluation, we trained all models on a fixed subset of the data.

	Recruiting Dataset			
	Rank Resume		Rank Job	
Method	Recall@100	nDCG@100	Recall@10	nDCG@10
ConFit*	79.00	44.21	80.83	60.67
Job2Resume	80.92	46.01	85.00	65.12
Resume2Job	74.18	42.30	77.75	59.82

Table A2: Comparison of generating hypothetical job posts instead of resume.

	Recruiting Dataset			
	Rank Resume		Rank Job	
Method	Recall@100	nDCG@100	Recall@10	nDCG@10
ConFit*	79.00	44.21	80.83	60.67
GPT-4o-mini (Few-shot)	80.92	46.01	85.00	65.12
Qwen (SFT)	71.62	42.07	78.50	57.38
Qwen (SFT+DPO)	77.17	45.07	84.75	63.53

Table A3: Comparison of Few-shot Prompting and Fine-tuning HyRe trained on subset

We present the results in Table A1. Compared to ConFit*, all three strategies show that HyRe yields a high potential performance if the hypothetical resumes can truly approximate the real selected ones. This indicates that HyRe, if implemented well, could significantly improve ranking job or ranking resume performance.

B.3 Hypothetical Job Embedding

In ConFit v2, we augment job data with hypothetical resumes generated by an LLM (Section 3.2). Alternatively, one can also augment resume data with hypothetical jobs generated by an LLM, which we experiment in this section.

We follow the same setup as Appendix B.2, but instead generate a hypothetical job for each resume. We then compare this against generating hypothetical resumes (HYRE). We present the results in Table A2. We find, on average, augmenting job post with hypothetical resume is more effective. We believe this is because job data are often short and contain less details compared to resumes, making it more beneficial for content expansion.

B.4 Finetuning HyRe

In this section, we examine whether fine-tuned LLMs are capable of generating hypothetical resume to approxiate Appendix B.2. We perform Supervised Fine-tuning (SFT) and Direct Preference Optimization (DPO) on Qwen-2.5 series models (Team, 2024). Specifically, we train Qwen-2.5-32B-Instruct with LoRA (Hu et al., 2022) on the Recruiting Dataset. All the experiments are implemented on 4 A100 80GB GPUs.

For SFT training, we select 6k high-quality resume-job pairs that have both a high human label score and a high cosine similarity score computed by Jina-v2' raw embeddings. For DPO training, we select the resume with the highest human label score as the chosen sample, and randomly select a resume with lower label score as the rejected sample. This results in a "preference" dataset with around 13k pairs. We then train Qwen-2.5-32B-Instruct with SFT and DPO, and evaluate the checkpoints after each training stage.

To speed up evaluation, we follow Appendix B.2 and evaluate all methods using a subset of the Recruiting dataset. We present the result in Table A3. We find after SFT+DPO training. the Qwen model improves CONFIT*, but it does not overpass Few-shot prompting on GPT-4o-mini. We believe this is because accepting a resume is subjective, making it hard to learn an "ideal candidate" directly from the dataset itself.

C Details on Dataset and Preprocessing

Recruiting Dataset The talent-job pairs are provided by a hiring solution company. The original resumes/job posts are parsed into text fields using techniques such as OCR. Some of the information is further corrected by humans. All sensitive information, such as names, contacts, college names, and company names, has been either removed or converted into numeric IDs. Example resume and job post are shown in Table A8 and Table A9, respectively.

Train	Recruiting Dataset	Aliyun Dataset
# Jobs	9469	19542
# Resumes	41279	2718
# Labels	48096	22124
(# accept)	19042	10185
(# reject)	29054	11939
# Token per R	1303.5(±947.3)	250.6(±96.5)
# Token per J	639.4(±1548.7)	335.1(±143.9)

Table A4: Training dataset statistics. "# Token per R/J" represent the *mean(±std)* number of token per resume/job after post-processing.

	Recruiting Dataset		AliYun Dataset	
Test	Rank R	Rank J	Rank R	Rank J
# Samples	200	200	300	300
# Jobs	200	300	300	2903
# Resumes	3000	200	1006	300

Table A5: Test dataset statistics.

AliYun Dataset The 2019 Alibaba job-resume intelligent matching competition provided resume-job data that is already desensitized and parsed into a collection of text fields. There are 12 fields in a resume (Table A8) and 11 fields in a job post (Table A9) used during training/validation/testing. Sensitive fields such as "居住城市" (living city) were already converted into numeric IDs. "工作经验" (work experience) was processed into a list of keywords. Overall, the average length of a resume or a job post in the AliYun dataset is much shorter than that of the Recruiting dataset (see Table A4).

We present the training and test dataset statistics in Table A4 and Table A5, respectively.

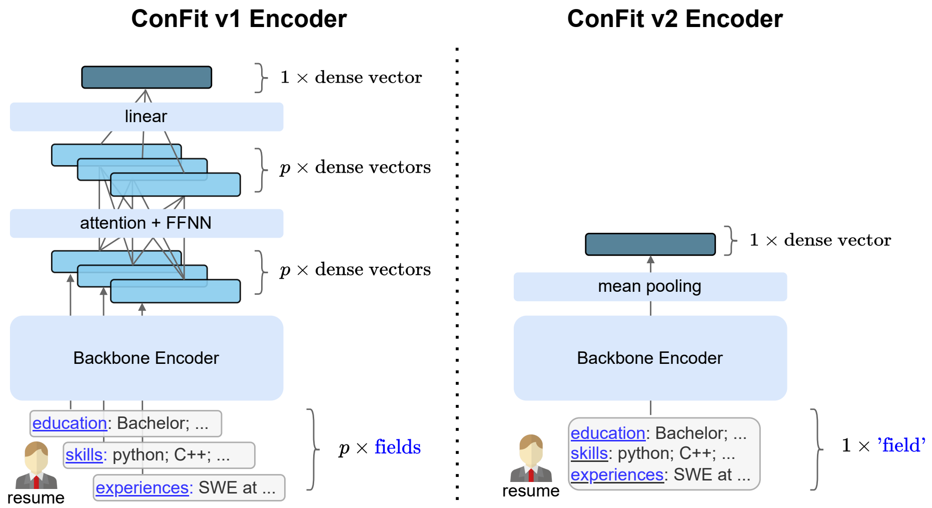

Figure A1: Encoder architecture comparison between ConFit (left) and ConFit v2 (right).

		Recruiting Dataset			
		Rank Resume		Rank Job	
Method	**Encoder**	nDCG@10-rand	nDCG@10-hard	nDCG@10-rand	nDCG@10-hard
RawEmbed.	E5-base	21.56	15.78	39.02	29.06
	Jina-base	11.36	12.42	34.28	34.30
	text-embedding-003-large	39.05	30.43	72.02	60.89
BM25	-	27.01	15.47	44.77	33.76
ConFit	Jina-base	33.65	24.57	65.67	54.67
ConFit* + RUM (ours)	Jina-base	49.96	39.92	84.05	73.69

Table A6: Comparing ranking performance of various approaches when test set consists of hard samples mined by BM25. We denote test sets using rejected and random unlabeled samples as negatives (Yu et al., 2024b) as "*-rand*", and test sets using rejected and unlabeled samples mined by BM25 as negatives as "*-hard*". Since test sets are smaller, we use nDCG@10 for all cases.

HyRe Prompt Template

Here is a template pair of matching resume and job:
[The start of the example job]
{{{example job}}}
[The end of the example job]
[The start of the example resume]
{{{example resume}}}
[The end of the example resume]
You are a helpful assistant. Following the above example pair of job and resume, construct an ideal resume for the target job shown below. You should strictly follow the format of the given pairs, make sure the resume you give perfectly matches the target job, and directly return your answer in plain text.
[The start of the target job]
{{{target job}}}
[The end of the target job]

Table A7: Few-shot prompt template for hypothetical reference resume generation. "*{{{...}}}*" are placeholders to be programmatically inserted during inference. "target job" is the job post to be augmented.

R **from Recruiting Dataset**	R **from AliYun Dataset**
## languages	## 期望工作城市
MANDARIN, ENGLISH	551,-,-
## industries	## 学历
IT.Electronic Industry	大专
## job functions	## 期望工作行业
SOFTWARE_AND_MATHEMATICS	房地产/建筑/建材/工程
## experiences	## 期望工作类型
most recent: Software Engineer for 8 years...	工程造价/预结算
second most recent: ...	## 当前工作行业
	房地产/建筑/建材/工程

Table A8: Example resume from the Recruiting dataset and AliYun dataset. The Recruiting dataset contains resumes in both English and Chinese, while the AliYun dataset contains resumes only in Chinese. All documents are available as a collection of text fields, and are formatted into a single string as shown above for ConFit* training.

J **from Recruiting Dataset**	J **from AliYun Dataset**
## Basic Info	## 工作名称
title: Sr. SWE-Perception Infra	工程预算
job type: Full-Time	## 工作城市
openings: 1	719
## job functions	## 工作类型
SOFTWARE_AND_MATHEMATICS	工程造价/预结算
## Requirements	## 最低学历
Strong programming skills in C++	大专
Minimum of a Masters Degree in CS or equivalent	## 工作描述
2+ years of experience in software industry...	能够独立完成土建专业施工预算...

Table A9: Example job post from the Recruiting dataset and AliYun dataset. The Recruiting dataset contains job posts in both English and Chinese, while the AliYun dataset contains job posts only in Chinese. All documents are available as a collection of text fields, and are formatted into a single string as shown above for ConFit* training.