MHSNet: An MoE-based Hierarchical Semantic Representation Network for Accurate Duplicate Resume Detection with Large Language Model

Yu Li*
Hangzhou Dianzi University
Hangzhou, China
liyucomp@hdu.edu.cn

Zulong Chen*
Alibaba Group
Hangzhou, China
zulong.czl@alibaba-inc.com

Wenjian Xu
Zhejiang University of Science and Technology
Hangzhou, China
wenjian.xwj@zust.edu.cn

Hong Wen†
Alibaba Group
Hangzhou, China
dreamonewh@gmail.com

Yipeng Yu
Taotian, Alibaba Group
Hangzhou, China
linxin.yyp@alibaba-inc.com

Man Lung Yiu
Department of Computing, Hong Kong Polytechnic University
Hong Kong, China
csmlyiu@comp.polyu.edu.hk

Yuyu Yin†
Hangzhou Dianzi University
Hangzhou, China
yinyuyu@hdu.edu.cn

Abstract

To maintain the company's talent pool, recruiters need to continuously search for resumes from third-party websites (e.g., LinkedIn, Indeed). However, fetched resumes are often incomplete and inaccurate. To improve the quality of third-party resumes and enrich the company's talent pool, it is essential to conduct duplication detection between the fetched resumes and those already in the company's talent pool. Such duplication detection is challenging due to the semantic complexity, structural heterogeneity, and information incompleteness of resume texts. To this end, we propose MHSNet, an multi-level identity verification framework that fine-tunes BGE-M3 using contrastive learning. With the fine-tuned BGE-M3, MHSNet generates multi-level sparse and dense representations for resumes, enabling the computation of corresponding multi-level semantic similarities. Moreover, the state-aware Mixture-of-Experts (MoE) is employed in MHSNet to handle diverse incomplete resumes. Experimental results verify the effectiveness of MHSNet.

CCS Concepts

• **Computing methodologies** → **Learning paradigms**; **Lexical semantics**; • **Information systems** → **Similarity measures**.

*Both authors contributed equally to this research.
†Corresponding Author: Yuyu Yin, Hong Wen.

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761547

Keywords

Duplicate Resume Detection, MoE, Semantic Representation

ACM Reference Format:
Yu Li, Zulong Chen, Wenjian Xu, Hong Wen, Yipeng Yu, Man Lung Yiu, and Yuyu Yin. 2025. MHSNet: An MoE-based Hierarchical Semantic Representation Network for Accurate Duplicate Resume Detection with Large Language Model. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3746252.3761547

1 Introduction

Duplicate Resume detection refers to the process of determining whether two given resumes belong to the same individual. This is challenging because those resumes may have missing information or inconsistent wording in their descriptions.

As shown in Figure 1, when recruiters are looking for candidates, they may scan and search from third-party websites (e.g., LinkedIn, Indeed). After obtaining the resumes from the third-party websites, recruiters will match them with the company's internal resume pool to identify duplicate resumes and merge the duplicate resumes into a more complete resume for subsequent job matching. However, resumes provided by third parties are often incomplete, and the same candidate may also upload multiple different resumes at different times. According to our statistics over a random sample of 6,000 resumes, we found that 99% contained inaccurate names, 1.5% lacked educational information, and 13% were with incomplete work experience. Duplicate detection in resumes with incomplete information plays a pivotal role in corporate recruitment. Given that companies conduct hundreds of thousands of interviews annually (via campus and social recruitment channels), each requiring substantial human and material resources, implementing resume

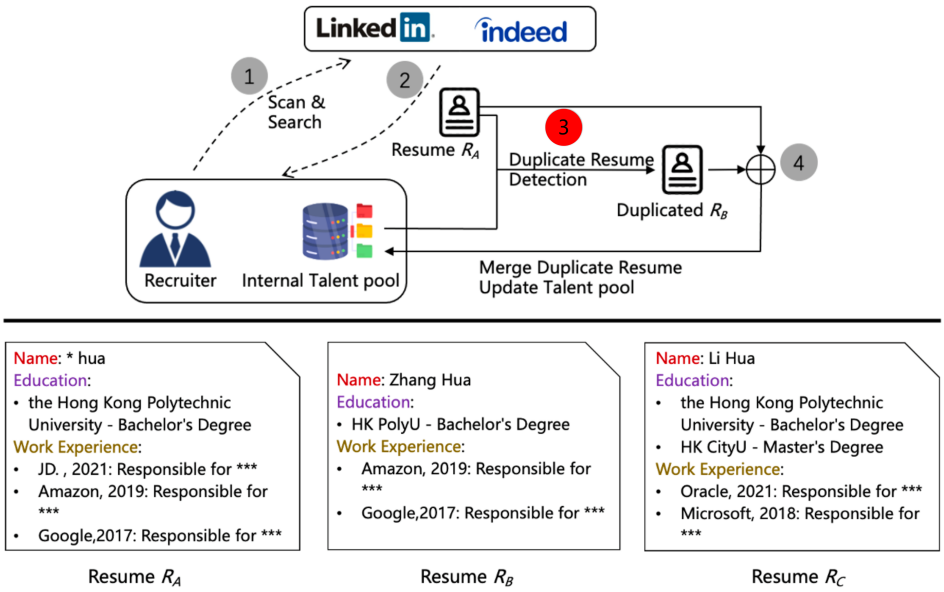

Figure 1: Example case of duplicate resume detection. Resumes R_A and R_B are duplicate while R_A and R_C represent different candidates.

duplication detection can save millions in operational costs while significantly enhancing recruitment efficiency.

Although natural language processing has been employed to improve resume-job matching [4, 13, 18, 23], current work primarily focuses on matching resumes with job requirements, which cannot be used to address the problem of resume de-duplication.

Duplicate resume detection is complex. As shown in Figure 1, although resumes R_A and R_C have the same undergraduate school in their educational background, their work experience do not match (e.g., JD in 2021 vs. Oracle in 2021). As comparison, although the school names in resumes R_A and R_B appear different, domain knowledge reveals that *HK PolyU* is simply an abbreviation of *the Hong Kong Polytechnic University*. Although the work experiences in R_A and R_B are not the same, it can be seen that R_B is a resume registered a few years ago, and thus the work experience in 2021 is missing. Therefore, R_A and R_B refer to the same candidate, meaning that they are duplicate resumes.

There are many challenges in duplicate resume detection: C_1, resumes are highly semantic-rich documents, making their semantic representation crucial. Additionally, resume data is imbalance, with negative sample (e.g., R_A and R_C in Figure 1) being particularly rare; C_2, resumes are typically long, semi-structured texts comprising numerous fields, such as educational and work experience. Both local and global similarities play a crucial role in duplicate detection; C_3, different resumes may have varying missing fields, making it challenging for a single network to handle all types of missing information effectively.

Therefore, to conduct accurate duplicate resume detection for recruiters, this paper proposes an multi-level duplicate detection framework MHSNet. In detail, to address C_1, MHSNet employs contrastive learning and element-wise data augmentation techniques to fine-tune embedding model BGE-M3 [2]. To address C_2, MHSNet calculates similarities between structured, semi-structured and full resumes to capture both global and local similarities between resumes. To address C_3, MHSNet utilizes Mixture-of-Experts (MoE) with the gating network directing the data to different expert modules according to the structured fields state in resumes. The main contributions could be summarized as follows:

- To our best knowledge, we are the first to conduct duplication detection over incomplete resumes.
- We propose MHSNet that leverages LLM, semantic computation, and MoE to conduct duplicate detection.
- We conduct extensive experiments on real-world resume datasets, and the experimental results show the superiority of our model.

2 Method

2.1 Problem Statement and System Overview

Definition 1. *Duplicate Resume Detection Problem* Given two resumes R_A, R_B, the duplicate resume detection problem is to calculate the similarities between R_A and R_B, so as to identify whether R_A and R_B belong to the same person. Specifically, a resume $R_A = \{R_A^{SG}, R_A^{SE}, R_A^{SO}, R_A^{C}\}$, where $R_A^{SG}, R_A^{SE}, R_A^{SO}$ are structured general, education, and occupation information, respectively. R_A^C is a set of semi-structured chunks which are split from the details of involved projects, received awards, and so on. Structured information refer to the content entered by users within predefined fixed modules of the resume template, while semi-structured information consist of text in open-ended areas of the template that allow flexible input.

As illustrated in Figure 2, to conduct duplicate resume detection, we propose MHSNet, an MoE-based Hierarchical Semantic Representation Network. Firstly, to deal with the semantic nature of resumes, MHSNet fine-tunes BGE-M3 [2] to achieve more accurate embedding. Specifically, element masking strategies are utilized to

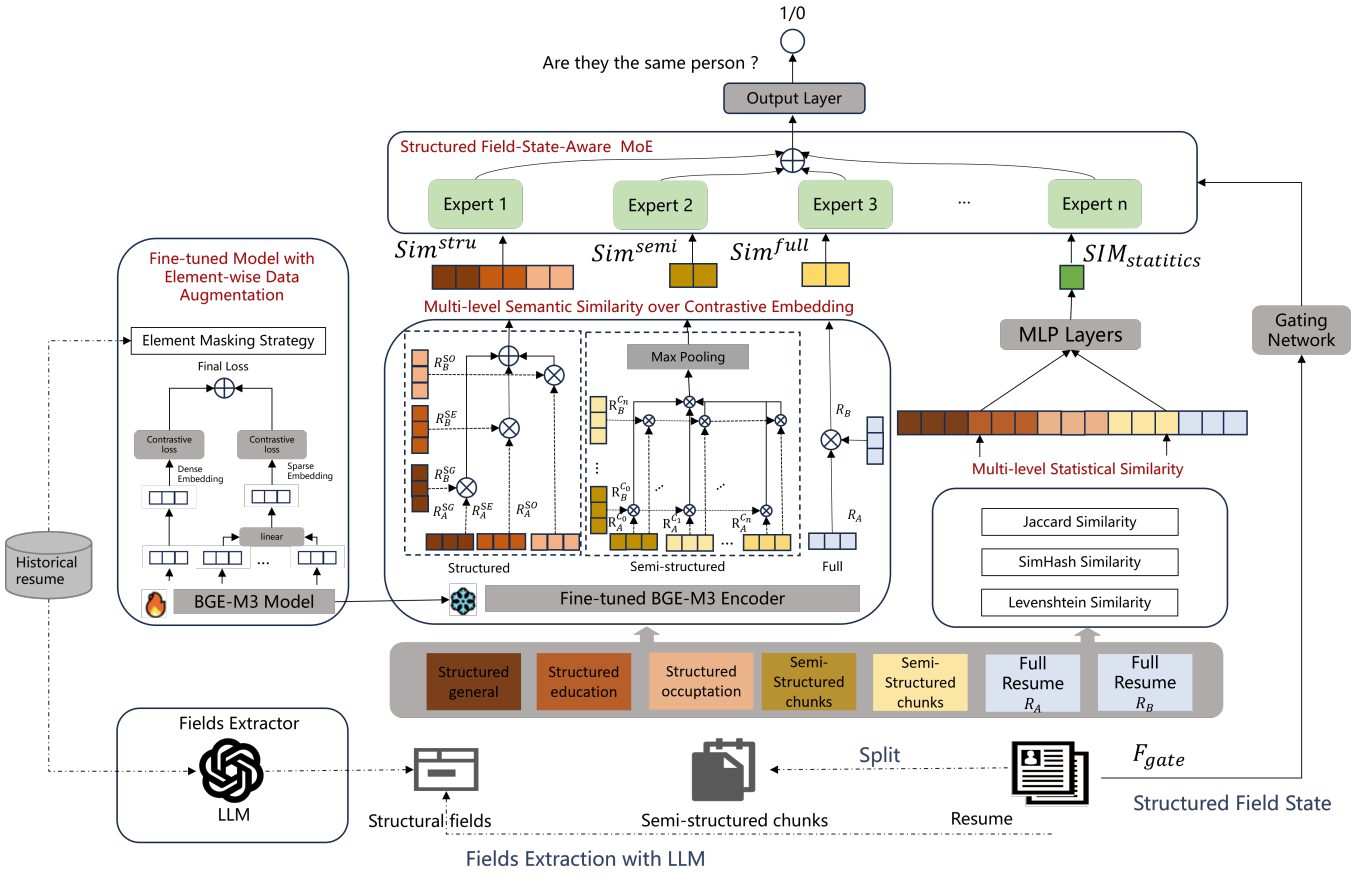

Figure 2: System architecture of MHSNet.

augment samples for better fine-tuning. Secondly, to fully utilize the local and global information in resumes, multi-level semantic similarity over contrastive embedding is computed. In detail, the fine-tuned BGE-M3 is used to get the embeddings for structured, semi-structured and full resumes. And the cosine similarity is calculated over these contrastive embeddings. Thirdly, since any embedding may lead to information loss, in order to preserve the original information in resumes, multi-level statistical similarity is computed. Finally, to deal with diverse missing fields, a state-aware MOE is proposed with a gating network to assign weights to each experts according to the input resumes.

2.2 Fine-tuned Model with Elements Data Augmentation

MHSNet fine-tunes BGE-M3 for embedding to achieve better resume representation. To do this, we first generate more positive and negative sample resumes to enrich the data, and then fine-tuning BGE-M3 with contrastive learning.

Elements Masking Strategy is proposed to generate more positive and negative sample resumes for fine-tuning. Specifically, we generate positive sample resumes through hiding part information in the original resume, and the masking strategy follows the distribution of resumes obtained from third-party. For negative samples, we generate resume pairs of different individuals, and these selected resumes have some similar structured information, and some similar semi-structured detailed experiences.

BGE-M3 Fine Tuning To capture semantics in resumes, we introduce rich semantic representations, including both dense and sparse representations. Moreover, in the face of the sparsity of samples and the limited number of positive and negative samples, we employ contrastive learning to fine-tune the model.

Specifically, the **dense embedding** v is used to capture the meaning of individual tokens and their interrelationships within a sentence. On the basis of the dense embedding, a learned **sparse embedding** v' is obtained through a linear layer, and is used for more precise matching and semantic-level similarity retrieval.

Contrastive learning is utilized to fine-tune BGE-M3 by learning the similarities and differences between samples. And the **loss for contrastive learning** depends on whether the samples are positive or not. In detail,

$$\begin{aligned} \mathcal{L}_{pos} &= (1 - cs)^2 \\ \mathcal{L}_{neg} &= reLU(0, (cs - 0.2)^2) \end{aligned} \tag{1}$$

where the *cs* represents the cosine similarity over the embedding vectors of two resumes. The fine-tuning process can enhance the robustness and the performance of BGE-M3.

2.3 Multi-Level Semantic Similarity over Contrastive Embedding

To capture local and global semantic similarities between resumes, we first extract structured fields from resumes with LLM, and split the semi-structured content in resumes into chunks. Then, for each resume, like R_A, multi-level sparse and dense contrastive embeddings are extracted with fine-tuned BGE-M3, including sparse and dense embeddings of structured fields R_A^{SG},R_A^{SE},R_A^{SO}, embeddings of semi-structured chunks $R_A^{C_i}$, and embeddings of full resumes R_A. After that, similarity networks are constructed to compute the multi-level semantic similarities. And the details of similarity networks are shown below.

As only information in the same structured field is meaningful to be compared, the **structured similarity network** computes the cosine similarity of the corresponding structured fields between R_A and R_B. Moreover, to capture rich semantic information, both sparse embedding and dense embedding are used in structured similarity network. As a result, the semantic similarity vector of structured fields is sim^{stru} as shown in Equation 2.

$$\begin{aligned} sim^{stru} &= concat(\mathcal{CS}^D_{SG}, \mathcal{CS}^S_{SG}, \mathcal{CS}^D_{SE}, \\ &\qquad \mathcal{CS}^S_{SE}, \mathcal{CS}^D_{SO}, \mathcal{CS}^S_{SO}) \\ \mathcal{CS}_{SG} &= \frac{R_A^{SG} \cdot R_B^{SG}}{\|\mathbf{R_A^{SG}}\|\|\mathbf{R_B^{SG}}\|} \end{aligned} \tag{2}$$

where $\mathcal{CS}^D_{SG}, \mathcal{CS}^S_{SG}$ represents the cosine similarity of the structured general information over dense and sparse contrastive embeddings, respectively.

Duplicate resumes should have some similar detailed experiences, like project experiences and award experiences. As the detailed experiences are semi-structured content and are always too long, we split the content into chunks according to specified rules (like fixed length or time intervals). Because incomplete resumes may lack details of some experiences, the highest similarity between semi-structured chunks is most likely to reflect the real similarity between two resumes. Thus, the **semi-structured similarity network** computes the semantic similarity between each pair of chunks in R_A and R_B, and the maximum value is identified through max pooling, as shown in Equation 3.

$$\begin{aligned} sim^{semi} &= concat(\mathcal{CS}^D_{semi}, \mathcal{CS}^S_{semi}) \\ \mathcal{CS}_{semi} &= \max_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \left(\frac{R_A^{C_i} \cdot R_B^{C_j}}{\|\mathbf{R_A^{C_i}}\|\|\mathbf{R_B^{C_j}}\|}\right) \end{aligned} \tag{3}$$

where $\mathcal{CS}^D_{semi}$ represents the maximum cosine similarity among dense-embedded semi-structured chunk pairs in R_A and R_B, while $\mathcal{CS}^S_{semi}$ represents the similarity over sparse embeddings.

Besides the local similarity computed in sim^{stru} and sim^{semi}, the global similarity between resumes is also important. Thus, the **unstructured similarity network** calculate the cosine similarity between full R_A and R_B, and the final semantic similarity $\mathbf{SIM}_{semantic}$ is obtained using Equation 4.

$$\begin{aligned} sim^{full} &= concat(\mathcal{CS}^D_{full}, \mathcal{CS}^S_{full}) \\ \mathcal{CS}_{full} &= \frac{R_A \cdot R_B}{\|\mathbf{R_A}\|\|\mathbf{R_B}\|} \\ \mathbf{SIM}_{semantic} &= concat(sim^{stru}, sim^{semi}, sim^{full}) \end{aligned} \tag{4}$$

2.4 Multi-Level Statistical Similarity

Although fined-tuned BGE-M3 can provide better embeddings to capture rich semantic information in resumes, any embedding may lead to information loss, which in turn affects similarity calculation. Therefore, we also compute the statistical similarities $\mathbf{SIM}_{statistic}$ between R_A and R_B using the complete original resumes. As shown in Figure 2, similar to $\mathbf{SIM}_{semantic}$, multi-level similarities are calculated over structured pairs, semi-structured pairs, and full pairs of R_A and R_B. Specifically, the Jaccard similarity, SimHash similarity and Levenshtein similarity are utilized as the measures to get a similarity vector $sim^{text} \in \mathbb{R}^{15}$. In order to more comprehensively represent the statistical similarity, an MLP layer is used to calculate as:

$$\mathbf{SIM}_{statistic} = \texttt{MLP}(sim^{text}) \tag{5}$$

2.5 Structured Field-State-Aware Mixture-of-Experts

After obtaining $\mathbf{SIM}_{semantic}$ and $\mathbf{SIM}_{statistic}$, MHSNet conduct duplicate detection between R_A and R_B. The impact of similarity between different pairs on final duplicate detection is influenced by the extent of information missing in the original resume. To better handle diverse missing information, we employ Mixture of Experts (MoE) to process semantic similarity and statistical similarity.

Specifically, according to Equation 6, the **gating network** takes the status of structured information in the resume as input. If the resume R_A lacks educational information, the variable $R_A^{SE}.isNULL$ is set to 1; otherwise, it is 0.

$$\begin{aligned} \mathcal{F}_{gate} &= linear(R_A^{SG}.isNULL, R_B^{SG}.isNULL \\ &\qquad R_A^{SE}.isNULL, R_B^{SE}.isNULL \\ &\qquad R_A^{SO}.isNULL, R_B^{SO}.isNULL) \\ \mathcal{G} &= softmax(\mathcal{F}_{gate}) \end{aligned} \tag{6}$$

In the **state-aware MOE**, each expert is an MLP network which independently trains and processes the input feature vector to obtain a corresponding score E_i, and the total score is calculated as the weighted sum of E_i according to Equation 7.

$$\begin{aligned} M_i &= \texttt{MLP}(concate(\mathbf{SIM}_{semantic}, \mathbf{SIM}_{statistic})) \\ Score &= \sum_{i=0}^{n} \mathcal{G}_i \cdot M_i \end{aligned} \tag{7}$$

2.6 Output Layer

The obtained *Score* in MOE is fed into the Output Layer, which ultimately outputs the duplicate resume detection result.

$$isDuplicate = softmax(Score) \tag{8}$$

3 Experiment

3.1 Experiment Settings

Datasets. We conduct the experiments on a real-world dataset collected by a large company in China. The dataset is collected for more than 6 months. There are 183205 resume pairs in the dataset, in which 160127 are positive samples and 23375 are negative samples, where positive samples refer to resumes that are textually dissimilar but belong to the same individual, while negative samples denote resumes with textual similarities that originate from different individuals.

Evaluation Metrics. We evaluate the duplicate detection task with three widely used metrics for classification: Area Under the Receiver Operating Characteristic (AUC), Accuracy, and F1 Score. Specifically, AUC is a curve drawn with true positive rate as the ordinate and false positive rate as the abscissa according to a series of different two classification methods (boundary value or decision threshold). AUC denotes the Area Under the receiver operating characteristic curve over the test set, which is a widely used metric for CTR prediction. The larger AUC is, the better the duplicate detection prediction model performs.

Baselines. To provide a comprehensive evaluation of our MHSNet model, we compare it against both traditional similarity calculation methods (i.e., Jaccard, SimHash and Levenshtein similarity) and LLM-based methods (i.e., mGTE [22], ME5 [17], BGE-Base [19] and BGE-M3 [2]).

Implementation Details. We adopted the BGE-M3 [2] as the LLM model in this paper. The training process is divided into two main parts: fine-tuning MHSNet and subsequent model training. During fine-tuning, given that it already possesses strong text representation capabilities, we conducted only one epoch of training with a batch size of 8 and a learning rate of $1e^{-5}$. Moreover, the dense embedding is set to $v \in \mathbb{R}^{250002}$ and the sparse embedding is set to $v' \in \mathbb{R}^{1024}$. After fine-tuning, we froze BGE-M3 and proceeded with the subsequent training phase. During detection model training, we trained for 20 epochs with a batch size of 1024 and a learning rate of $1e^{-3}$. Additionally, in our model design, each gate consists of a linear layer, and each expert is composed of a 3 layer MLP.

3.2 Evaluation Results

Overall Performance. As shown in Table 1, our proposed MHSNet outperforms all baseline solution, verifying the effectiveness of MHSNet. Among the baseline models, BGE-M3 generally outperform traditional similarity calculation methods, while BGE-M3(dense+sparse) with supervised contrastive learning perform better than those trained with unsupervised learning. This suggests that models fine-tuned with contrastive learning can effectively compute text similarity, leading to more accurate resume similarity judgments. Moreover, among all fine-tuned models, BGE-M3 performed exceptionally well, which is the main reason for choosing BGE-M3 for fine-tuning.

Table 1: Performance of the proposed and baseline methods for duplicate resume detection, where $*$ indicates the best result among baselines. Improvement refers to the enhancement achieved by MHSNet.

Models	Accuracy	AUC	F1
Jaccard	0.5036	0.5532	0.5036
SimHash	0.4460	0.6025	0.4296
Levenshtein	0.7050	0.4265	0.8194
ME5-base	0.7554	0.7973	0.8211
BGE-Base-zh-v1.5	0.7122	0.7871	0.7727
mGTE-Base	0.7770	0.8048	0.8394
BGE-M3(only dense)	0.7410	0.8317	0.8043
BGE-M3(only sparse)	0.8417	0.8486	0.8922
BGE-M3(dense+sparse)	0.8489*	0.8593*	0.8976*
Ours MHSNet	**0.8849**	**0.9096**	**0.9223**
Improvement	**5.13%**	**7.19%**	**3.37%**

Ablation Studies. As depicted in Table 2, all of the components in our proposed MHSNet have significant impacts on the performance of duplicate resume detection. In detail, removing sim^{semi} (Equation 3), sim^{full} (Equation 4), and Statistical similarity (Equation 5) has a minor effect on the MHSNet performance, with accuracy decreasing by only about 1%. However, after removing sim^{stru} (Equation 2) accuracy drops from 88% to 83%, suggesting that basic information, educational background, and work experience play a crucial role in resume duplicate detection. Compared to Dense embeddings, Sparse embeddings, through the added linear layer that calculates token weights in the text, play a critical role in resume duplicate detection, particularly in structured comparison. Furthermore, the accuracy of the model without fine-tuning dropped by 6%, while fine-tuning significantly improved the accuracy, confirming the effectiveness of our contrastive learning-based fine-tuning approach. Finally, after removing the MOE component, accuracy decreased by 3%, indicating that MOE plays a key role in determining duplicate resume detection.

Table 2: Performance of the Ablation Studies.

Model	Accuracy	AUC	F1
w/o Dense embedding	0.8345	0.9059	0.8844
w/o Sparse embedding	0.7985	0.8911	0.8494
w/o Fine-tuning	0.8201	0.8893	0.8663
w/o sim^{stru}	0.8345	0.9024	0.8832
w/o sim^{semi}	0.8705	0.8937	0.9108
w/o sim^{full}	0.8705	0.8979	0.9126
w/o Statistical similarity	0.8776	0.9091	0.9154
w/o MoE	0.8561	0.8945	0.9009
MHSNet	**0.8849**	**0.9096**	**0.9223**

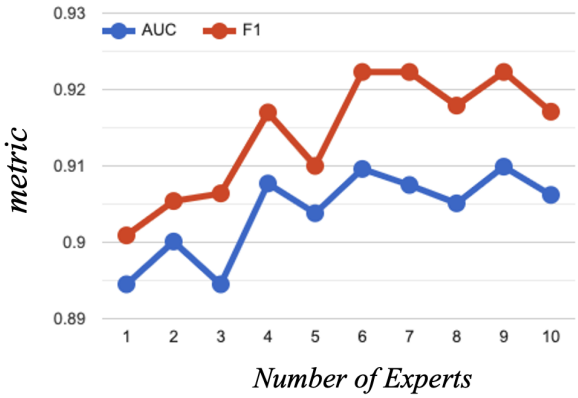

Figure 3: Effect of Number of Experts.

Effect of Chunk Division Methods. Table 3 illustrates that constructing semi-structured chunks according to time intervals can achieve better performance, as this division method can better capture the temporal dependencies in the sequence.

Table 3: Effect of Chunk Division Methods

Chunk Division Mode	AUC	F1
Fixed length(length=512)	0.9064	0.9162
Divided by line break	0.9069	0.9154
Divided by time interval	**0.9096**	**0.9223**

Effect of Number of Experts. Figure 3 investigates the impact of the number of experts on the judgment results. We evaluate the performance of MHSNet with varying numbers of experts, from 1 to 10. The experimental results show that when the number of experts is 6, the model performs well over both AUC and F1 metrics. Therefore, we adopted this setting in subsequent experiments.

Online A/B Test. Figure 4 illustrate the pipeline of online system of duplicate resume detection model (i.e., Step 3 in Figure 1). The online system is deployed on Alibaba's original platform and mainly consists of an online service module and an offline training/inference module. The system leverages a search engine platform to retrieve similar resumes, then uploads the features of these resumes to the online service system, where candidate resumes are re-ranked to determine whether the input resume has duplicates within the internal resume pool.

It is conducted over the real world system used in a large company in China. We compare the duplicate resume detection performance between the original system and the updated system with MHSNet. To ensure fairness, we adjusted the scheduling engine on the online platform so that, during the online A/B tests, approximately half of the daily traffic for duplicate resume detection is allocated to each model. We monitor the online result for 8 days, and record the accumulated bad cases as shown in Figure 5. Here, a bad case refer to a negative sample, which means the system considers two resumes to be from the same person, but in fact, they

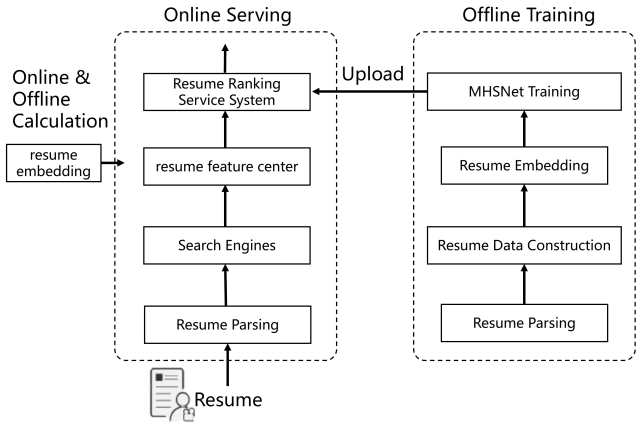

Figure 4: The Duplicate Resume Detection Model in Online System.

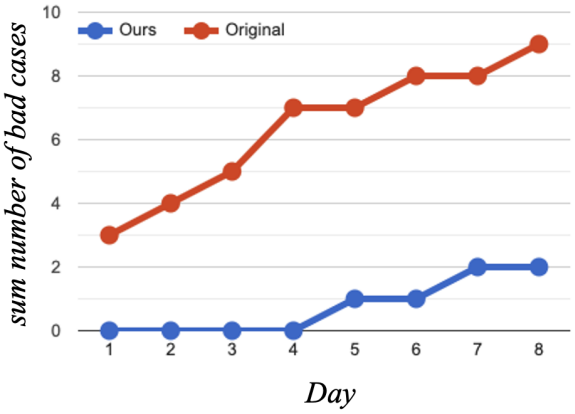

Figure 5: Online A/B test.

Table 4: Case Study of BGE-M3 Fine Tuning

Text in Structured Field	no Fine-tuning		with Fine-tuning	
	CS_S^D	CS_S^S	CS_S^D	CS_S^S
James Michael Smith J. M. Smith	0.7353	0.5365	0.9751	0.9732
CityU Hong Kong City University	0.8686	0.5157	0.9526	0.9488
ByteDance TikTok	0.4907	0.5005	0.6118	0.5

are not. According to the result, MHSNet can improve the online performance. Moreover, the results also reveal the sparsity and skewness of real-world data, specifically, the scarcity of negative samples.

Case Study of Verifying Fine-tuning. A case is depicted in Table 4 where the similarity in each row refers to CS_{SG}^D (Line 1 in Table 4), CS_{SE}^D (Line 2) and CS_{SO}^D (Line 3). The result verify that fine-tuning BGE-M3 is necessary. The fine-tuned model indicate significant improvements in both Dense and Sparse scores, with

Name: James *
Education:
- Carnegie Mellon University - Bachelor's Degree
- Stanford University - Master's Degree

Work Experience:
- Twitter,2021:Responsible for the A/B Testing Platform enhancement initiative
- Google,2019:Responsible for the development of the Search Query Understanding System.

Resume A

Name: James *
Education:
- Carnegie Mellon University - Bachelor's Degree
- Stanford University - Master's Degree

Work Experience:
- Twitter,2021:Responsible for improving the performance of the Tweet Ranking Algorithm.
- Uber,2018:Responsible for optimizing the Dynamic Pricing System across global markets.

Resume B

Figure 6: Example Resumes.

the Sparse score nearly doubling from 0.5365 and 0.5157 to 0.9732 and 0.9488, respectively. Even for completely dissimilar name text pairs, the fine-tuned model's dense score increased by about 20%. These results clearly indicate that contrastive learning fine-tuning significantly enhances the model's ability to capture implicit semantic relationships in resumes, such as name similarity (full names and abbreviation), school and company abbreviation normalization, and other multi-level semantic similarities, thereby improving the overall performance of the model.

Case Study for Verifying MOE. Considering the resumes R_A and R_C in Figure 1, the corresponding $\mathcal{F}_{gate} = [0, 1, 1, 1, 1, 1]$ as the name field in R_A is incomplete. In this case, $\mathcal{G} = [0.0131, 0.8710, 0.8538, 0.8487, 0.4553, 0.3084]$, and the final weighted score is 0.3597, which is lower than the score of 0.5325 without using MOE. This indicates that MOE can deal with various missing types well.

Case Study for sim^{stru}. We analyze the effectiveness of the structured information network sim^{stru}(Equation 2). Figure 6 presents similar resume cases for two different job applicants, where both Resume A and Resume B use fictitious names. Both resumes have the same education experience and worked at the same company in 2021, but in different specific roles. To calculate sim^{stru}, we first obtain the General similarity between Resume A and B as 0.95, the Education similarity as 0.9964, and the Occupation similarity as 0.2521. With sim^{stru}, the overall similarity is calculated as 0.3597, which is significantly lower than the similarity of 0.7607 without sim^{stru}. The result indicates that the structured information network can effectively extract the similarity of names and educational background, while reducing the similarity score for subtle differences in work experience, thus more accurately distinguishing between the two resumes.

4 Related Work

Job recommendation has been widely studied [9, 14]. Text-based methods were introduced, to encode job descriptions and resumes[13, 15, 24] . Behavior-based methods can capture complex user-job interactions [5, 21]. Hybrid models combined both text and behavior data [6, 7, 10]. Recent trends include adversarial training, behavior graphs [1, 11], and LLM-based recommendations [4, 18].

Semantic Similarity is widely used in text classification[8] and machine translation[25]. Transformer-based models such as BERT[3] have introduced contextual understanding by combining pre-training and fine-tuning. In addition, advances in text embedding models(e.g., E5 [16], BGE [20], and Sentence-T5 (ST5) [12]) have improved semantic similarity measurement.Furthermore, to meet the demands of multilingual scenarios, models such as BGE-M3 [2] have been developed.

5 Conclusion

We provide a thorough analysis of the challenges in duplicate resume detection, with particular emphasis on the structural heterogeneity of resumes (structured, semi-structured, and unstructured components). To address the challenges of duplicate resume detection, we propose MHSNet, a novel framework for detecting duplicates in incomplete resumes. The key strength of MHSNet lies in its hierarchical processing of diverse resume structures, ensuring robust adaptation to heterogeneous input formats. Furthermore, the integration of Mixture of Experts (MOE) enables specialized processing of distinct resume fields. In detail, MHSNet first fine-tunes BGE-M3 with contrastive learning and element-wise data augmentation. Then, local and global similarities between resumes are calculated over different part of resumes. After that, MOE is utilized to handle diverse incomplete cases. Effectiveness of MHSNet is verified with comprehensive experiments.

Acknowledgments

This work is supported in part by the Natural Science Foundation of Zhejiang University of Science and Technology (No. 2025QN023), the Zhejiang Province Key R&D Program (No. 2023C01217), the Fundamental Research Funds for the Provincial Universities of Zhejiang (No. GK249909299001-017), the Natural Science Foundation of Zhejiang Province (No. LQ24F020040), the Graduate Course Development Project of Zhejiang University of Science and Technology (No. 2024yjskj03), the Ideological and Political Education Teaching Research Project of Zhejiang University of Science and Technology (No. 2024-ksj3), and Yangtze River Delta Science and Technology Innovation Community Joint Research Project (No. 2022CSJGG1000/2023ZY1068).

GenAI Usage Disclosure

The authors hereby disclose that no generative AI technologies were used in the creation or writing of this manuscript.

References

[1] 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. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 65–74.

[2] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[4] Yingpeng Du, Di Luo, Rui Yan, Xiaopei Wang, Hongzhi Liu, Hengshu Zhu, Yang Song, and Jie Zhang. 2024. Enhancing Job Recommendation through LLM-Based Generative Adversarial Networks. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (Eds.). AAAI Press, 8363–8371.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[6] Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Shuqing Bian, Yang Song, Tao Zhang, and Ji-Rong Wen. 2022. Leveraging search history for improving person-job fit. In *International Conference on Database Systems for Advanced Applications*. Springer, 38–54.

[7] Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning effective representations for person-job fit by feature fusion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2549–2556.

[8] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *CoRR* abs/1408.5882 (2014). arXiv:1408.5882 http://arxiv.org/abs/1408.5882

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[10] Ran Le, Wenpeng Hu, Yang Song, Tao Zhang, Dongyan Zhao, and Rui Yan. 2019. Towards effective and interpretable person-job fitting. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1883–1892.

[11] Yong Luo, Huaizheng Zhang, Yonggang Wen, and Xinwen Zhang. 2019. Resumegan: an optimized deep representation learning framework for talent-job fit via adversarial learning. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1101–1110.

[12] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pretrained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1864–1874. doi:10.18653/v1/2022.findings-acl.146

[13] 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*. 25–34.

[14] Rohan Ramanath, Hakan Inan, Gungor Polatkan, Bo Hu, Qi Guo, Cagri Ozcaglar, Xianren Wu, Krishnaram Kenthapadi, and Sahin Cem Geyik. 2018. Towards deep and representation learning for talent search at linkedin. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2253–2261.

[15] et al. Shen. 2018. Joint Representation Learning for Person-Job Fit. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

[16] 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. *arXiv preprint arXiv:2212.03533* (2022).

[17] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv:2402.05672 [cs.CL] https://arxiv.org/abs/2402.05672

[18] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2024. Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9178–9186.

[19] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[20] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 641–649. doi:10.1145/3626772.3657878

[21] 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 *Proceedings of the 16th ACM Conference on Recommender Systems*. 102–112.

[22] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. arXiv:2407.19669 [cs.CL] https://arxiv.org/abs/2407.19669

[23] Zhi Zheng, Zhaopeng Qiu, Xiao Hu, Likang Wu, Hengshu Zhu, and Hui Xiong. 2023. Generative Job Recommendations with Large Language Model. *CoRR* abs/2307.02157 (2023). arXiv:2307.02157 https://doi.org/10.48550/arXiv.2307.02157

[24] 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. *ACM Transactions on Management Information Systems (TMIS)* 9, 3 (2018), 1–17.

[25] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1393–1398.