DOI: https://doi.org/10.15276/aait.07.2024.27
UDC: 004.89

ResJobFit - end-to-end artificial neural networks based technology for job-resume matching

Maiia Y.Bocharova[1)]
ORCID: https://orcid.org/orcid=0009-0004-3875-5019; bocharova.maiia@gmail.com. Scopus Author ID: 57193357730
Eugene V. Malakhov[1)]
ORCID: https://orcid.org/0000-0002-9314-6062; eugene.malakhov@onu.edu.ua. Scopus Author ID: 56905389000
[1)] Odesa I. I. Mechnikov National University, 2, Dvoryanska Str. Odesa, 65082, Ukraine

ABSTRACT

With the ever-growing expansion of online recruitment, a reliable person-job matching has become increasingly crucial. Due to different experience, education and specialization requirements, as well as location considerations, specified in the job advertisement, various aspects should be taken into account for reliable matching and ranking of candidates. It has been shown that matching of resumes and vacancies can be approached as either pair classification or semantic similarity search based on embeddings. While classification approaches process each vacancy-resume pair sequentially, thus resulting in quadratic time complexity, independent text embeddings and ranking is a much more efficient and scalable solution, since it has linear time complexity. In this article semantic similarity search to rank suitability of candidates with regards to vacancies has been used. ResJobFit - an end-to-end Artificial Neural Networks based technology for job-resume matching is proposed. ResJobFit technology consists of Segmentation, Parsing, Summarization and HR Embedding Module models, and their outputs (vector and attributes defining each resume or job advertisement), as well as a Vector Database in which the records are stored. Unsupervised text embeddings training for HR domain encapsulating two novel training objectives - intra- and cross-section contrastive alignment is introduced. Pretrained BERT-base model is adapted by teaching it to match summary-last employment sections of the resume with parts of the same vacancy or employment section. As baselines TFIDF, BERT, E5 and GTE have been used. The proposed unsupervised training strategy was compared against SimCSE, DeCLUTR and ConFit approaches. NDCG, MAP and MRR are used as metrics for measuring accuracy of the designed algorithm. It has been shown that the novel training objective lets it achieve significant improvement in comparison to other unsupervised training approaches. Improvement of 11% in NDCG was achieved by adapting the DeCLUTR training strategy for the HR domain based on exploiting the structure of resumes over the classical DeCLUTR training strategy on the task of ranking summaries of vacancies and resumes generated by large language models. 2% and 6% have been achieved using ResJobFit and ResJobFit with requirements matching over state-of-the-art ConFit model on the task of ranking full-text vacancies and resumes.

Keywords: Artificial neural networks; IT systems; machine learning; NLP; transformers; text embedding; information retrieval

For citation: **Bocharova M. Y., Malakhov E. V. "ResJobFit - end-to-end Artificial Neural Networks based technology for job-resume matching".** ***Applied Aspects of Information Technology*****. 2024; Vol.7 No.4: 378–391. DOI: https://doi.org/10.15276/aait.07.2024.27**

INTRODUCTION

The recruitment process is a fundamental aspect of human resource management. The primary goal of recruitment is to hire candidates who bring value to the company. This process is viewed from two perspectives: recruiters and job seekers. Recruiters create job descriptions by outlining the necessary skills, expertise levels, and qualifications; while job seekers prepare their resumes by detailing their educational background, work experience, and skills.

In the rapidly changing landscape of today's job market, organizations face significant challenges in acquiring top talent efficiently and effectively. The intensifying "war for talent" and the overwhelming volume of applications for open positions have introduced new complexities in processing and selecting candidate profiles. Traditionally, HR departments have managed the recruitment process by manually evaluating resumes and inviting candidates for interviews based on their qualifications. This process involves screening resumes to determine if a candidate meets the job's requirements and subsequently conducting interviews to assess both their skills and personality fit for the role.

However, for large companies and mass recruiters, managing vast amounts of candidate data and interview details has become increasingly burdensome. The conventional approach, which heavily relies on manual resume screening and basic keyword matching, is fraught with several limitations:

– Inefficiency: Manually reviewing a large number of resumes is not only time-consuming but also prone to human error, leading to the possibility of overlooking qualified candidates;

– Subjectivity and discrimination: Human judgment in resume screening can introduce biases, potentially disadvantaging candidates based on gender, demographic, racial factors and leading to unfair evaluations.

The rapid advancement of information technology (IT) and artificial intelligence algorithms has revolutionized various sectors [1], and recruitment is no exception. More and more researchers are focusing on automatic person-job matching, proposing various approaches to enhance the efficiency and accuracy of the process.

The integration of Natural Language Processing (NLP) in recruitment not only expedites the hiring process but also minimizes human biases, leading to a fairer selection procedure. As the demand for efficient and effective recruitment continues to grow, the role of NLP in transforming traditional recruitment practices becomes increasingly critical.

Generally, automatic resume-job matching and recommendations fall under one of the four categories:

– Content-based ones - which use a semantic similarity measure between job advertisement and resume representations. In the early days those systems relied on keyword-based matching techniques [2], later approaches employ word2vec techniques [3] and BERT-like [4] encoder models and either encode the full resume [5], or use BERT representations of sentences and produce sentence expression matrix, which is then pulled to obtain the final representation of the resume or job advertisement [6].

Those approaches however suffer from not being able to explicitly take into account all the mandatory requirements like comparing the years of work experience, education level and its major.

– Collaborative filtering – those approaches use behavioral data. Usually for this type of systems training data is collected using the candidate-job click interaction recording. As such, authors of [7] propose to recommend jobs to the user based on which jobs similar users have interacted with. [8] use auto-encoders to encode interaction’s of the candidates with the vacancies, based on which the similar candidates are identified. Such approaches require the interaction data and typically suffer from the cold start problem, when the candidate has not done any interaction with job posts yet.

– Hybrid – in those ones a combination of several models is used. Many studies employ BERT-like pre-trained language models or CNNs and add hierarchical attention networks. For example, in [9] use a combination of recommendations based on interaction history of the candidate along with the similarity-based recommendations. In [10] authors created different models and used a multi-stack ensemble to combine them.

– Knowledge-based – using some predefined taxonomy, both candidate’s resume and job posting are classified into one of the classes of the ontology, and afterwards the matching process is conducted. As such, in [11] authors propose to use an occupation category classifier before ranking the jobs to reduce the search space. In spite of the substantial efforts required to create such ontologies, this is the approach which is used by the leading companies like LinkedIn and Daxtra Technologies.

Combining the content-based and ontology-based methods promises to exploit the strengths of both and create a more robust matching system. Thus, a content-based model can handle the nuances of language and the context within resumes and job descriptions, while ontology-based filtering can guarantee that specific industry standards and requirements are met, while ensuring the explainability of the results.

It must also be taken into consideration that resumes are submitted in various formats and formattings, as such correctly parsing and structuring information poses a significant problem on its own. Most approaches overlook the parsing step completely, assuming that data comes in the structured format or make an assumption that data is already prepared for use. This gap necessitates the development of robust systems that can handle raw, unstructured data effectively, ensuring accurate and reliable person-job matching.

Therefore, research in the area of end-to-end resume-job matching is relevant.

1. ANALYSIS OF LITERARY DATA

With the advent of Artificial Intelligence (AI), Deep Neural Networks and spread of adoption of Language Models recruitment processes have undergone significant transformations, offering new possibilities for efficiency and effectiveness [12, 13].

Historically submitted resumes were reviewed and analyzed manually by recruiters [14]. However, as big companies often have to deal with large volumes of submitted resumes, automating candidate review and shortlisting can save time and costs and improve the overall efficiency of the recruitment process.

1.1. Text embeddings

The encoding model’s capability is of utmost importance for the quality of the results matched by the retrieval system. Thanks to the introduction of large-scale pretrained language models like BERT [3], and methods for adapting general-purpose models for the task of creating meaningful text representation like SBERT [15] or RetroMAE [16]. Methods leveraging large scale datasets with text pairs annotated for similarity have made it possible to create high-performance models, however in many scenarios labeled data is not available or difficult to obtain. This made the main focus of the recent research focus on the development of unsupervised learning methods.

As such, some approaches use auto-encoding based training strategies and encoder-decoder architecture. Most well-known are TSDAE [17], which adds noise by swapping, adding and deleting words and after encoding the noisy text trains decoder to reconstruct the original text. RetroMAE [16] also uses encoder-decoder architecture, in which the input text passed to the encoder is masked at a low ratio and [CLS] token representation along with the original text where 50-70% of the tokens are masked is passed to the decoder for reconstruction, and encoder and decoder parts are trained simultaneously. Although effective, those strategies require training a separate decoder model which leads to additional resource consumption during training.

Another group of approaches leverage recent advances in large language models (LLMs). These models, trained on diverse and extensive corpora, exhibit a profound understanding of language structure and semantics and are able to follow instructions and generate high-quality synthetic data. As such, in [18] authors suggest to use GPT-3 to generate training triplets for text embedding learning. However, this approach is expensive, as is involves utilizing high-cost proprietary APIs. Additionally, the synthetic data generated may not fully correspond to the target domain's specific language, style, or nuances, potentially reducing the relevance and effectiveness of the resulting embeddings in specialized applications.

Alternatively, self-contrastive learning, in which positive samples are constructed from the unlabeled text using either some augmentation techniques or co-occurring text fragments within a larger context has become increasingly popular among researchers. In this paradigm, after constructing positive pairs, contrastive loss is utilized to train the model to discriminate the positive samples from the in-batch negative ones. Methodologies of construction of positive pairs are an active research area.

As such, in “SimCSE” [19] authors use different dropout masks applied to the same text for construction of positive samples. The obvious drawback of such an approach lies in the fact that the positive pairs are constructed from exactly the same text, with identical words and length. This limits the diversity of information between positive pairs, as no additional semantic variations or paraphrasing are introduced. Consequently, the model is prone to overfitting to surface-level patterns and struggle to generalize to scenarios where semantically similar texts differ in wording, structure, or length.

Authors of [20] use neighboring text chunks as positive pairs, along with a large batch size. Similarly, in “DeCLUTR” [21] authors propose to create positive pairs leveraging non-overlapping text chunks from the same document. This approach has the advantage of introducing semantic diversity in positive pairs, as different parts of the same document usually provide complementary or contextually related information. However, while demonstrating effective results, when drawing text chunks randomly, semantic similarity between them can vary greatly, leading to noisy or weak positive pairs that could hinder model performance.

Therefore, the importance of developing efficient positive sample selection approaches for unsupervised contrastive learning of text representations cannot be overestimated.

1.2 Resume-Job advertisement matching

In the swiftly changing job market of today, acquiring talent efficiently and effectively has become a crucial challenge for companies. The intense competition for top talent, coupled with the high volume of applications for available positions, has introduced new complexities in processing and selecting candidate profiles [22]

The resume-vacancy matching systems model the suitability between resume and job advertisement, allowing to find the most suitable candidates for the position. A resume (or job advertisement) is usually a semi-structured document, which exceeds the 512 tokens limit imposed by many transformer-based model architectures like BERT. This token limitation poses a challenge for effectively capturing the complete semantic information from such lengthy documents.

In [5] authors take the first 512 tokens of the document as the model’s input and train the siamese network model on more than 274 thousand of labeled resume-vacancy pairs. The training process involves optimizing both classification and regression objectives to improve the model's performance. This however can lead to suboptimal performance due to information loss.

Similar limitations face authors of [23], who restrict in the study to only the last employment history of the candidate, fully ignoring all other details present in the resume. Subsequently they train a model for sequence pair classification task - with positive pairs constructed by using combinations of job responsibilities which a person had, while negative pairs comprise pairs of responsibilities coming from different people’s profiles. During inference the trained model is fed by both vacancy and last employment history of the candidate and the classification score is used to evaluate the similarity between them. Restricting to only considering the last employment section can severely hinder the model’s performance, as it does not account for education and certification requirements. This limitation is particularly pronounced for recent graduates who may not have had a professional job aligned with their field of study or career aspirations. Many graduates take on part-time or unrelated jobs during their university years, which may not reflect their qualifications, skills, or potential.

There are some approaches, which strive to address this issue by embedding sentences or text chunks separately and then aggregating the representations.

In [24] authors train an adversarial network to reconstruct the job post by its vector representation, and the resume is split into three parts - namely talent experiences, talent skills and other factors, which are concatenated and passed through MLP layers to obtain final resume representation. After that representations of resume and job post along with their element-wise product are concatenated and used to make a binary prediction about candidate’s suitability for the job. Such comparisons however suffer from quadratic computational complexity, as the approach requires evaluating every resume against every job post in the dataset.

In the most recently introduced ConFit [6] authors strive to solve some of the shortcomings mentioned above and propose an augmentation approach for resume and job advertisement representation modeling. They propose to use augmentations of the section’s text - both EDA (Easy Data Augmentation) techniques and LLM-based paraphrasing to generate the positive samples and then proceed to train an encoding model using contrastive learning with in-batch negative samples. However, their approach relies on initial labeled resume-job advertisement pairs dataset and only employs augmentation to increase the number of samples.

As such, while showing good performance on structured and preprocessed data, existing methodologies usually fully overlook the preprocessing steps. Apart from that, conventional approach of framing the resume-job matching problem as a binary classification task introduces significant computational overhead due to the quadratic complexity of comparing all possible vacancy-resume pairs. Lastly, and perhaps most notably, there are currently no unsupervised training approaches for text modeling specifically designed for the HR domain, leaving a significant gap in leveraging domain-specific knowledge for more effective and efficient candidate-job matching.

2. THE PURPOSE AND OBJECTIVES OF THE RESEARCH

The purpose of this research is improving ranking metrics for matching jobs with resumes, while maintaining linear time complexity for the matching process.

To achieve the defined goal, tt is necessary to address the following problems:

– compile an evaluation dataset for measuring models’ performance;

– adapt unsupervised training strategy specifically suitable for HR-related text;

– develop a generic resume and job advertisement matching technology that can be applied to different formats of both types of documents and which can handle resumes and vacancies which exceed the 512 token limit imposed by BERT transformer architecture.

3. RESEARCH METHODS

In order to address the challenges of efficiently (with linear time complexity) parsing and matching resumes with vacancies, we propose ResJobFit – an end-to-end technology designed to streamline this process.

ResJobFit extracts and matches mandatory requirements such as locations, educational degrees, number of years of experience. After this step dense vectors produced by the Machine Learning system are compared, which allows ResJobFit to quickly filter and match candidates to the respective vacancy.

3.1. Substantiation of the methodology for developing the end-to-end resume-job matching technology

The technology of ResJobFit involves several critical steps to ensure accurate and efficient processing, which are described in detail below.

The full schema of resume-job matching can be presented as shown in Fig. 1.

As can be seen from Fig. 1, ResJobFit technology consists of Segmentation, Parsing, Summarization and HR Embedding modules, and their outputs (vector and attributes defining each resume or job advertisement), as well as a Vector Database (DB) in which the records are stored.

A. Segmentation stage involves section-based segmenting the text into a predefined set of categories for future processing. For Resume task this predefined set of categories includes “zone_title”, “personal_info”, “skills_summary”, “employment”, “project”, “education”, “certification”, “training”, “publication”, “reference”. Vacancy is segmented into 3 categories: “company_and_benefits”, “profile” and “job” category. Texts are segmented using the styles and capitalization aware modification of the BERT model, as was described previously [25].

As a result of segmentation, contents of the resume are divided into three major super-sections, which highlight the different aspects of the candidate’s profile:

– objective, summary and skills – this group captures the self-description of the candidate, their skills and career aspirations;

– employment – this group provides a comprehensive overview of the candidate’s past job roles, responsibilities, and achievements;

– education, training, license and certifications – this group includes academic qualifications and any pertinent formal training and licenses which the demonstrate the resume's qualifications.

B. Parsing stage is necessary to extract important structured attributes. Among them are start and end dates of employment, educational degrees and certifications. For parsing CapStyleBERT architecture was used as described previously in [25].

C. Summarization stage plays an important role in providing recruiters with a concise overview of the candidate, as well as highlighting the most relevant details for the matching process. Summarization process is organized as shown in Fig. 2.

Phi-3 was fine-tuned on GPT-4o generated summaries [26]

D. HR Embedding Module is needed to transform textual data into fixed-sized vectors suitable for comparisons. The schema of this module is presented in Fig. 3.

Each super-section is designed to provide a focused view of different aspects of a candidate’s profile, though it is recognized that some resumes or vacancies may lack specific components. For instance, certain candidates may not have any mentions of licenses or certifications. Similarly, recent graduates or entry-level candidates may not have had any previous employment.

Mathematically the above can be summarized with the following formulas:

$$h_{ij} = HR\_Embedder(t_{ij}), \qquad (1)$$

where h_{ij} is the embedding of j-th token of i-th supersection; t_{ij} is the j-th token in the i-th supersection

The representation of the i-th super-section, h_i, is obtained by taking the mean of its token embeddings:

$$E_i = \frac{1}{N}\sum_{j=1}^{N} h_{ij}, \qquad (2)$$

where N is the number of tokens in the i-th the supersection; h_{ij} is the embedding of j-th token of i-th supersection; E_i is the embedding of supersection.

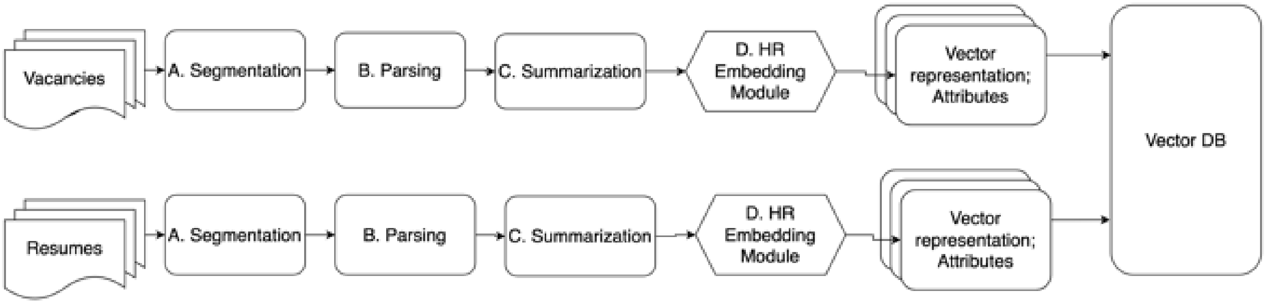

Fig. 1. **AI ResJobFit technology**
Source: **compiled by the authors**

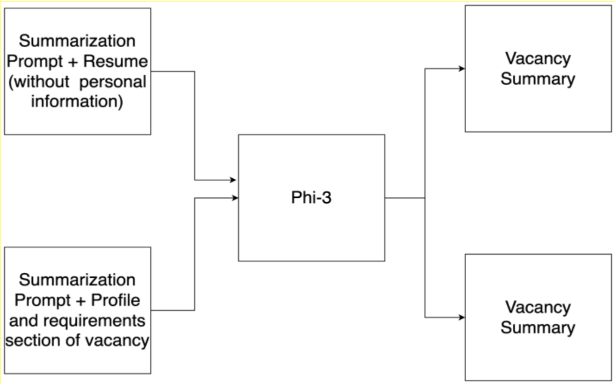
Summarization Prompt + Resume (without personal information)
Summarization Prompt + Profile and requirements section of vacancy
Phi-3
Vacancy Summary
Vacancy Summary

Fig. 2. **Schema of resume summarization pipeline organization**
Source: **compiled by the authors**

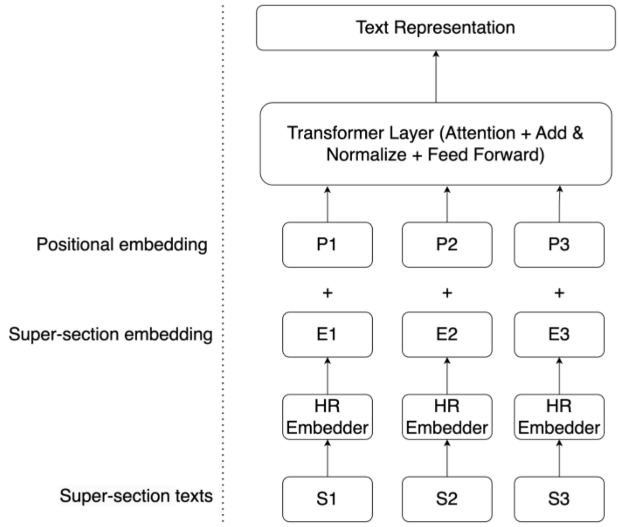

Fig. 3. **HR texts representation module**
Source: **compiled by the authors**

Afterwards each super-sections is encoded separately using a pre-trained language model which we define as HR Embedder. This step transforms the textual content of each text group into a sequence of embeddings, the mean of the token embeddings is taken as a section representation. This representation serves as a unique “token” in a higher-level sequence, with each super-sections token contributing to an overall profile representation.

Vacancy description is segmented into two parts – “job” - with information relative to future functions of the person and “profile” – description of qualification and requirements which are stated in the job ad.

To enable the model to create contextualized final representation, an attention mechanism along with absolute positional encoding is incorporated. This encoding helps the model retain the information about the order of sections, enabling it to capture both the hierarchical and contextual nuances.

The sequence $\{E_1, E_{2,...,} E_M\}$ is enriched with absolute positional encodings P_i to retain section order information:

$$z_i = E_i + P_i \qquad (3)$$

An attention mechanism processes the sequence $\{z_{1,}z_{2,...}z_M\}$ to generate a contextualized profile representation

$$E_{final} = Attention(\{z_{1,}z_{1,}...z_M\}), \qquad (4)$$

where E_{final} is the final resume or vacancy representation; M is number of supersection in the document (3 in resume and 2 in vacancy); z_i is contextualized representation of i-th supersection.

3.2. Unsupervised representation learning in the HR domain

Getting a large-scale annotated dataset with matched vacancy-resume pairs may not be feasible in most real-world scenarios, due to the labor-intensive and costly nature of creating such datasets. Manual annotation of resumes and vacancies requires domain expertise, which significantly increases the overhead. Moreover, the dynamic and evolving nature of job markets, with new roles and requirements emerging constantly, means that pre-existing labeled datasets can quickly become outdated. As a result, models trained on static, manually curated data may struggle to generalize to new job descriptions and resumes.

To address this issue, a synthetic data generation pipeline that leverages the most recent employment information from resumes can be used. By transforming the most recent employment information extracted from a real resume into job vacancy description through the use of large language models (LLMs), a high volume of positive resume-vacancy pairs can be created automatically. This approach not only bypasses the need for manually annotated datasets but also introduces greater flexibility and scalability, allowing the model to be trained on diverse, up-to-date examples.

Once the most recent employment data is extracted, it is passed through a large language model (LLM), such as GPT-4, to generate a synthetic job vacancy description that aligns with the role the candidate held. The LLM is tasked with transforming the resume’s employment details into a formal job posting format, simulating how an employer might describe a similar open position in their company.

The prompt provided to the LLM is structured to guide it in generating a comprehensive job vacancy description, capturing essential elements such as job responsibilities, required qualifications, and desired skills.

To avoid direct hints in the data, the final step involves removing the corresponding employment entry from the original resume. This ensures that the generated job description is not directly reflected in the resume, simulating a real-world scenario where a candidate applies for a similar role but their resume does not contain a perfect job match. By removing the employment entry, the synthetic pair becomes a valid training example for the model, mimicking the real-world task of matching resumes to job vacancies.

However, given that generating synthetic data using Large Language Models (LLMs) is resource-intensive, exploring fully unsupervised pretraining strategies that reduce required dataset size is essential. Human-written resumes provide a valuable, natural dataset for this purpose, as they contain a wealth of structured information and linguistic nuances across diverse job functions. This makes them an ideal pretraining resource for natural language processing tasks such as textual semantic similarity.

Building on the idea introduced in DeCLUTR [19], in which positive pairs are sampled as non-overlapping spans from the same document, this approach can be improved to utilize the inherent structure of resumes. Addressing the shortcoming of random selection of text spans, enhanced selection strategies can exploit the structured nature of resumes. For instance, the “Summary” section, which often provides an overarching view of an individual’s professional profile, can be paired with the most recent role’s description from the employment section. This leverages the natural alignment between high-level professional summaries and specific job responsibilities, ensuring that positive pairs are contextually rich and semantically coherent.

Similarly, within the employment section itself, task-oriented bullet points associated with a single job role can be grouped intelligently. Rather than random sampling, positive pairs can be formed by selecting complementary or sequential tasks that collectively describe the scope of the role. Specifically, sections describing a particular employment role often present a list of task-oriented lines which describe the role and responsibilities which the individual executed. Those lines share a context of one job, describing different aspects of the role performed. For example, a person can write “Vacuuming and cleaning carpets” and “Laundry; Making beds” in one job description, highlighting different aspects of one job performed. Thus, randomly selecting such tasks from one job description to form a positive pair can force the model to learn nuanced representations that capture the essence of specific job functions. This enhances its understanding of skills and duties associations within particular roles and reinforces contextual alignment within roles.

The “Summary” section, which is usually closely aligned with the individual’s most recent employment role, linked with the lines describing actual tasks of the last employment section can provide another source of positive pairs. Summaries often encapsulate key skills, accomplishments, and responsibilities, presenting a concise reflection of the candidate's primary strengths and areas of expertise. When lines from the summary are paired with lines from the employment section, they form strong, contextually aligned positive pairs that reinforce the model's capacity to capture core competencies relevant to the individual’s professional profile. This approach allows the model to generalize better across resumes with different structures and styles, building a richer understanding of how roles and skills are presented. By incorporating both task-specific and high-level descriptions in contrastive learning, the model can develop representations that are not only contextually grounded within specific job functions but also broadly applicable across various employment scenarios, supporting improved accuracy in job-resume matching.

In the contrastive training paradigma negative samples are selected from the same batch. Specifically, all in-batch samples except for the positive one for a specific sample are used as negatives for it. This forces the model to learn discriminative features and to distinguish between semantically related and random texts.

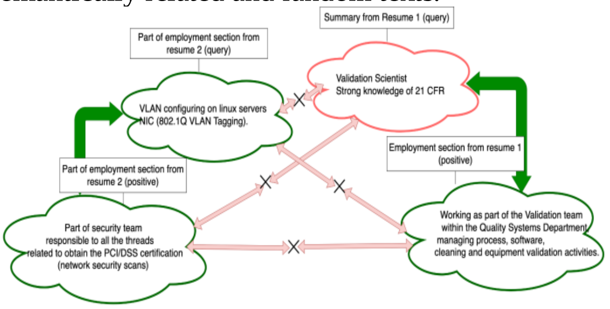
Summary from Resume 1 (query)
Part of employment section from resume 2 (query)
Validation Scientist
Strong knowledge of 21 CFR
VLAN configuring on linux servers
NIC (802.1Q VLAN Tagging).
Employment section from resume 1 (positive)
Part of employment section from resume 2 (positive)
Part of security team responsible to all the threads related to obtain the PCI/DSS certification (network security scans)
Working as part of the Validation team within the Quality Systems Department managing process, software, cleaning and equipment validation activities.

Fig. 5. **Unsupervised pretraining of text embedding model**

***Source:* compiled by the authors**

As a similarity score – cosine similarity is selected due to its ability to measure the angular distance between vector representations in high-dimensional spaces regardless of the magnitude of vectors. In line with the research findings, loss is calculated in a bidirectional manner.

As a similarity score – cosine similarity is selected due to its ability to measure the angular distance between vector representations in high-dimensional spaces regardless of the magnitude of vectors. In line with the research findings, loss is calculated in a bidirectional manner

$$Loss_{s_j,emp_j} = -log_e \frac{e^{sim(s_j,emp_j)}}{\sum_{i=1}^{k} e^{sim(s_i,emp_j)} + \sum_{i\neq j}^{k} e^{sim(s_i,s_j)} + \sum_{i\neq j}^{k} e^{sim(emp_i,emp_j)}}, (5)$$

where $sim(s_j, emp_j)$ is the cosine similarity of vectors s_j and emp_j; s_j is vector representation of the “summary” extracted from the resume which is in the j-th position in the batch; emp_j is vector representation of the “employment” section extracted from the resume which is in the j-th position in the batch; e is the base of the natural logarithm; log_e is the natural logarithm; k is the size of the batch; $Loss_{s_j,emp_j}$ is the loss relative to the specified index j in the batch.

$$Loss_{p1_j,p2_j} = -log_e \frac{e^{s(p1_j,p2_j)}}{\sum_{i=1}^{k} e^{s(p1_i,p2_j)} + \sum_{i\neq j}^{k} e^{s(p1_i,p1_j)} + \sum_{i\neq}^{k} e^{s(p2_i,p2_j)}}, (6)$$

where $s(p1_j, p2_j)$ is the cosine similarity of vectors $p1_j$ and $p2_j$; $p1_j$ is vector representation of a part of one position description (either employment section of resume or requirements descriptions section of vacancy), which is in the j-th position in the batch; $p2_j$ is vector representation of a different part of the sample position description which is in the j-th position in the batch; e is the base of the natural logarithm; loge is the natural logarithm; k is the size of the batch; $Loss_{p1_j,p2_j}$ is the loss relative to the specified index j in the batch.

Final loss is defined as follows:

$$Loss = \sum_{j=1}^{k} (Loss_{s_j,emp_j} + Loss_{p1_j,p2_j}), (7)$$

where $Loss$ is in the final loss for the batch of size k.

By calculating and maximizing during training the cosine similarity between the vectors of related text pairs and at the same time pushing representations of the unrelated (random) text pairs apart in vector space the model learns to efficiently compare HR texts.

4. METHODS AND MODELS

A large and diverse data, consisting of ~4 million resumes and ~1.4 million vacancies was used in experiments. Both resumes and vacancies were segmented as described in “3.1.1 Section-based segmentation”.

4.1. Statistics of the dataset used for fully unsupervised training

The dataset extracted from resumes included 20.4 million unique employment description sections. After filtering out entries with fewer than six bulleted lines, 3.6 million sections were retained for analysis. Each selected employment section was then randomly split into two parts without repetition, ensuring that each segment contained a minimum of three lines.

Parts defining requirements and profile-related parts of vacancy were prepocessed separately the same way as employment description sections of resumes. Such preprocessing made it possible to create 2.4 million positive pairs.

Summary sections were extracted from resumes and validated in terms of informativeness by checking the number of skill-phrases present. Records with non-informative summaries, which did not contain a title or at least 3 skills were filtered out. Only records from resumes which contained at least 5 lines in the employment section were considered. This way 996k unique summary-employment section pairs were constructed.

Because of the data imbalance, resume-employment-parts and vacancy-parts positives were downsampled, to match the volume of summary-employment positive pairs, ensuring balanced training with up to two iterations on the summary-employment pairs.

Final statistics of the dataset used for unsupervised training are shown in Table 1.

Table 1. **Statistics of the dataset used for unsupervised training**

Data Source	Number of samples	Average length (words)
Summary-employment pairs extracted from resume *(cross-section alignment)*	996k	94
“job” and “profile” parts of vacancy *(intra-section alignment)*	996k	61
employment section of resume *(intra-section alignment)*	996k	48

Source: **compiled by the authors**

4.2. Statistics of the synthetic dataset used for training if HR Embedding Module

One hundred thousand resumes which contain at least three employment sections are taken for experiments. Most recent employment section is used. Statistics of the training data are provided in Table 2.

Table 2. **Statistics of synthetic dataset for training HR embedding module**

Statistics	Average length (words)
Average number of words per Resume's "supersection 1" (objective, summary, skills)	146
Average number of words per Resume's "supersection 2" (2nd and 3rd most recent employments)	148
Average number of words per Resume's "supersection 3" (education, training, certification, publications)	53
Synthetic vacancy produced by LLM	232

Source: **compiled by the authors**

5. EXPERIMENTS

5.1. Unsupervised training of HR Embedder

Unsupervised training was conducted using multiple negative ranking loss as defined in Equation (3).

The truncation for HR embedder was set to 128 tokens as it was empirically found that further increasing the sequence length does not lead to noticeable improvement in performance, while posing higher computational requirements.

Performance of the model on validation set of real-world matched vacancies - resume pairs (summary-based) was measured during pretraining of HR Embedder.

Experiments were conducted on a T4 GPU, a batch size of 64 was employed for contrastive training of HR Embedder. The AdamW optimizer was employed, with a learning rate set to 2e-5 and a linear warmup phase applied over the first 10 % of training steps.

5.2. Unsupervised training of HR Embedding Module on synthetic dataset

Training was conducted with representations of resumes and synthetially generated vacancies with as defined in Equation (4) using Multiple Negative Ranking Loss.

Experiments were conducted on A100 GPU with the batch size set to 96. The AdamW optimizer was used, along with a learning rate set to 2e-5 and a linear warmup phase applied over the first 10% of training steps.

6. EXPERIMENTAL RESULTS

6.1. Compiling the evaluation dataset for measuring models' performance on HR texts and defining metrics

Human judgment is crucial for benchmarking resume recommendation systems, since they provide the gold standard for evaluating accuracy of automated systems

The evaluation of the proposed technology is conducted using a dataset constructed from real-world recruiters' data provided by a private online recruitment company.

To ensure high quality of the data, cleaning and debiasing methods are applied. As such, language detection is conducted by using an open-source FastText model and records with non-English content are excluded from the dataset.

The final evaluation dataset comprises 12,068 resumes and 200 vacancy descriptions, thus accurately reflecting the dynamic and varied nature of real-world job markets. To ensure robustness and generalizability, the dataset includes resumes and job descriptions from multiple industries and occupational sectors.

The dataset comprises two versions – full-text records and concise summaries of each entry, generated using a large language model (LLM). This dual representation allows for a more thorough evaluation of models across different levels of detail and abstraction. The full-text version mirrors real-world data, with an average length of 509 words per vacancy and 862 words per resume.

In contrast, the summary-based version offers much shorter entries – averaging 96 words per vacancy and 102 words per resume – making it suitable for benchmarking classical embedding models. The summary-based version is made publically available to facilitate further research in the area [27]

For the evaluation the proposed recommender system cosine-similarity as the final similarity metric to recommend resumes to a particular vacancy is used.

To address the challenge posed by the typical length of resumes and vacancies, which often exceed the 512-token limit of most BERT-like architectures, two scenarios were tested. The first scenario employs AI-generated summaries, which provide a condensed overview of the content to capture the essential information while reducing input length. The second one involves using original resumes and vacancies.

To assess the performance of the developed system, several standard metrics are employed:

– mean Average Precision (MAP): Measures the average proportion of correctly matched resumes to the total number of resumes retrieved for each vacancy. Number of resumes retrieved is set to the number of relevant resumes for each particular vacancy;

– mean Reciprocal Rank (MRR): Evaluates the ranking quality of the matched resumes by measuring the average of the reciprocal ranks of the first relevant resume match for each vacancy serving as a query;

– normalized Discounted Cumulative Gain (NDCG): measures the ranking quality. It’s a result of the division of Discounted Cumulative Gain (DCG) and ideal DCG. DCG is the gain (relevance) of each correctly retrieved resume in the ranking list, with a logarithmic discount applied based on the position of the particular result. Ideal DCG denotes the best DCG which can be achieved if all resumes are ranked correctly.

6.2 Comparative Analysis of performance of models trained without supervision for representing HR texts

To validate the effectiveness of the proposed method, a comparative analysis against existing state-of-the-art general text embedding approaches, as well as specific ones designed for resume-vacancy matching – ConFit [5] was conducted. Since the weights of the Confit model are not available, “BERT-base” is fine-tuned on the same data in accordance with the strategy described in the original ConFit paper (EDA and ChatGPT Augmentations), as well as the ResJobFit technology proposed in the paper for fair comparisons. Results of the experiments can be seen in Table 3.

In line with the findings of other researches, original BERT without specific fine-tuning performs worse than TFIDF (0.08 lower MAP score than TFIDF). While “SimCSE” demonstrates better performance than BERT, it still lags behind models trained with supervision on human-labeled data, highlighting the importance of tailored training strategies. “DeCLUTR” achieves slightly improved results over “SimCSE”, likely due to diverse positive samples seen during training when leveraging non-overlapping text spans from the same document. However, as can be seen from Table 2, random sampling of spans as described per “DeCLUTR” strategy, without leveraging the inherent structure of resumes, does not not provide enough semantic alignment of positive samples. In particular, the obtained results (11% NDCG improvement) underscore the importance of intelligent sampling of positive pairs which take into account the inherent structure of resumes.

Table 3. **Performance of the models on AI-generated summaries**

Model	MAP	MRR	NDCG
TFIDF	0.39	0.61	0.64
BERT-base	0.31	0.56	0.61
e5-base-v2	0.55	0.75	0.77
gte-base	0.56	0.75	0.78
SimCSE*	0.421	0.642	0.688
DeCLUTR*	48.93	0.681	0.728
ConFit	0.59	0.775	0.79
HR Embedder	0.61	0.79	0.81

Source: **compiled by the authors**

The proposed method surpasses even the “ConFit” strategy (2.5 % improvement) , which rely on augmentations generated by LLMs, while solely utilizing data extracted from resumes and job advertisements. This makes it both more cost-effective and scalable, as it eliminates the dependency on computationally expensive synthetic data generation while maintaining superior performance.

6.3. Evaluation of performance of ResJobFit technology

Performance of models on full-text resumes and vacancies is show in Table 4. As can be seen from it, effective strategies for segmenting and aggregation of resumes are crucial, and encoding resumes as they are leads to a substantial degradation of performance for general deep learning based approaches (“e5-base-v2” performance is 17 % worse when full-text resume is used compared to its performance when AI-generated summary).

Table 4. **Performance of the models on whole text resumes**

Model	MAP	MRR	NDCG
TFIDF	0.42	0.62	0.66
BERT-base	0.19	0.39	0.47
e5-base-v2	0.47	0.71	0.72
gte-base	0.52	0.73	0.76
ConFit	0.60	0.79	0.80
HR Embedding Module (no hard requirements filtering)	0.63	0.82	0.82
HR Embedding Module (with requirements filtering)	0.65	0.85	0.86

Source: **compiled by the authors**

The proposed ResJobFit technology outperforms these models, achieving higher metrics compared to both general text embedding models as well as ConFit (2 % improvement in NDCG compared to “ConFit” strategy), which also incorporated segmentation of resumes in the processing pipeline. Additionally, filtering based on hard requirements extracted from vacancies leads to further improvements (4% improvement compared to ResJobFit without requirements filtering). This can be attributed to the fact that such filtering helps mitigate the errors originating from specific seniority requirements like “5+ years of experience”.

In addition to quantitative metrics, qualitative evaluations were conducted by soliciting feedback from recruitment professionals. These professionals assessed the relevance and quality of the matches produced by the proposed system. The feedback indicated an improved satisfaction with the recommendation provided by the system, particularly noting the accuracy of skill and experience alignment between resumes and vacancies.

7. DISCUSSION OF OBTAINED RESULTS

In the stude, ResJobFit – a general-purpose end-to-end retrieval-oriented technology to model resume-job fit was proposed. This technology consists of text segmemtation, parsing, entity normalization, hard-requirements matching and deep-learning based reranking. A synthetic positive resume-vacancy generation pipeline is proposed to create training data for the deep learning matching model. To form positive pairs, the candidate’s most recent employment history is extracted from the resume and passed through a large language model (LLM) to generate a vacancy description that aligns with the role obtained by the person. The corresponding employment entry from the resume is removed. This approach alleviates the need for the often difficult and expensive to obtain annotated text pairs.

Unsupervised pretraining for human resource management related texts is proposed, which constitutes of using different parts of the same job description written by either recruiter or candidate for creation of high-quality positive pairs for contrastive pretraining.

ResJobFit is trained using a contrastive learning approach. Deep-learning based matching is tested in two scenarios: summary-based matching and whole text matching, in the later the sections of the resume are encoded independently and passed through an attention layer to get final resume representation. Experiments on the real-world data show the effectiveness of the ResJobFit against existing State-of-the-Art methods.

Limitations

While ResJobFit demonstrates strong performance in modeling resume-job fit, there are several limitations that should be acknowledged. First, the quality of the generated synthetic vacancy descriptions depends heavily on the capabilities of the large language model (LLM). Despite recent advancements, LLMs can introduce biases or inaccuracies when generating job descriptions based on the provided employment section, particularly for highly specialized or niche roles, which could affect the quality of the positive pairs.

Future research

ResJobFit demonstrates promising results on English language data, however adapting it to multi-language settings remains an open problem. LLMs used for generating vacancy descriptions are typically trained on predominantly English corpora, which may not capture the linguistic nuances or context of non-English job descriptions and resumes. Adapting ResJobFit to support multilingual input would require careful data curation, would involve not only translating resumes and vacancies but also addressing cultural and linguistic variations in how qualifications and job roles are described across different regions.

The model is trained with a contrastive learning framework that benefits from a well-defined set of positive and negative pairs. Although synthetic positive pairs are easy to get and multiple negative ranking loss shows good results when negative pairs are lacking, defining a strong negative mining approach can further improve the performance.

CONCLUSIONS

1. The evaluation dataset for measuring models' performance on resume-job advertisement matching task was compiled. This dataset consists of 200 vacancies and 12.068 resumes in two formats – full-text version and a summarized version generated by a large language model (LLM).

2. Intra – and cross-section alignment unsupervised training strategy for HR texts was introduced, which lead to 11 % improvement in NDCG compared to DeCLUTR strategy (summarized version of the dataset).

3. ResJobFit – a generic resume and job advertisement matching technology that can be applied to different formats of both types of documents was developed. It consists of Segmentation, Parsing, HR Embedder models and their outputs (vector and attributes defining each resume or job advertisement), which are stored in the Vector Database. This technology achieves linear time complexity for job advertisement-resume matching and 2 % and 6 % improvement in NDCG compared to "ConFit" strategy without and with hard requirements matching respectively. (full-text version of the dataset).

REFERENCES

1. Andronati, O., Antoshchuk, S., Babilunha, O., Arsirii, O., Nikolenko, A. & Mikhalev K. "A method of constructing ensemble classifiers for recognizing audio data of various nature". *14th International Conference on Advanced Computer Information Technologies (ACIT)*. Ceske Budejovice, Czech Republic. 2024. p. 758–761, https://www.scopus.com/authid/detail.uri?authorId=54419480900.
DOI: https://doi.org/10.1109/ACIT62333.2024.10712469.

2. Kumaran, V. S. & Sankar, A. "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping". *EXPERT. Int. J. Metadata Semant. Ontol.* 2013; 8 (1): 56–64. DOI: https://doi.org/10.1504/IJMSO.2013.054184.

3. Gugnani, A. & Misra, H. "Implicit skills extraction using document embedding and its use in job recommendation". *In Proceedings of the AAAI Conference on Artificial Intelligence*. 2020; 34: 13286–13293. DOI: https://doi.org/10.1609/aaai.v34i08.7038.

4. Devlin, J. "Bert: Pre-training of deep bidirectional transformers for language understanding". *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019. p. 4171–4186, https://www.scopus.com/authid/detail.uri?authorId=54879967400. DOI: 10.18653/v1/N19-1423.

5. Lavi, D., Medentsiy, V. & Graus, D. "conSultantBERT: Fine-tuned siamese sentence-bert for matching jobs and job seekers". 2021. DOI: https://doi.org/10.48550/arXiv.2109.06501.

6. Yu, X., Zhang, J. & Yu, Z. . "ConFit: Improving resume-job matching using data augmentation and contrastive learning". *Association for Computing Machinery*. 2024. p. 601–611. DOI: https://doi.org/10.1145/3640457.3688108.

7. Schmitt, T., Philippe, C. & Michele, S. "Matching jobs and resumes: a deep collaborative filtering task". *In Proceedings of the 2nd Global Conference on Artificial Intelligence*. 2016. p. 1–14.

8. Lacic, E., Reiter-Haas, M., Kowald, D., Reddy Dareddy, M., Cho, J. &, Lex, E. "Using autoencoders for session-based job recommendations". *User Model User-Adap Inter 30*. 2020. p. 617–658. DOI: https://doi.org/10.1007/s11257-020-09269-1.

9. Nigam, A., Roy, A., Singh, H. & Waila, H. "Job recommendation through progression of job selection". *In IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. 2019. p. 212–216. DOI: https://doi.org/10.1109/CCIS48116.2019.9073723

10. Carpi, T., Edemanti, M., Kamberoski, E., Sacchi, E., Cremonesi, P., Pagano, R. & Quadrana, M. "Multi-stack ensemble for job recommendation". *In Proceedings of the Recommender Systems Challenge.* 2016. p. 1–4. DOI: https://doi.org/10.1145/2987538.2987541.

11. Zaroor, A., Maree, M. & Sabha, M. N. "A hybrid approach to conceptual classification and ranking of resumes and their corresponding job posts". *KES International Conference on Intelligent Decision Technologies*. 2017. p. 107–119. DOI: https://doi.org/10.1007/978-3-319-59421-7_10.

12. Malakhov, E., Shchelkonogov, D. & Mezhuyev, V. "Algorithms for classification of mass problems of production subject domains". *In proceedings of 8th International Conference on Software and Computer Applications*. 2019. p. 149–153, https://www.scopus.com/authid/detail.uri?authorId=56905389000. DOI: https://doi.org/10.1145/3316615.3316676.

13. Bisikalo, O., Kovtun, O. &, Kovtun, V. “Neural network concept of ukrainian-language text embedding”. *13th International Conference on Advanced Computer Information Technologies (ACIT)*. Wrocław, Poland. 2023. p. 566–569, https://www.scopus.com/authid/detail.uri?authorId=57105837600. DOI: 10.1109/ACIT58437.2023.10275511.

14. Qin, C., Zhang, L., Cheng, Y., Zha, R., Shen, D., Zhang, Q. & Xiong, H. “A comprehensive survey of artificial intelligence techniques for talent analytics”. 2023. DOI: https://doi.org/0.48550/arXiv.2307.03195.

15. Reimers, N. & Gurevych, I. “SentenceBERT: Sentence embeddings using siamese BERT networks”. *Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong. 2019, https://www.scopus.com/authid/detail.uri?authorId=57028066100.
DOI: https://doi.org/10.18653/v1/D19-1410.

16. Xiao, S., Liu, Z., Shao, Y. & Cao, Z. “RetroMAE: Pre-Training Retrieval-oriented Language Models Via Masked Auto-Encoder”. *In proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, https://www.scopus.com/authid/detail.uri?authorId=57223161443. DOI: https://doi.org/10.18653/v1/2022.emnlp-main.35.

17. Wang, K., Reimers, N. & Gurevych, I. “Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning”. *In Findings of the Association for Computational Linguistics: EMNLP*. Punta Cana, Dominican Republic. Association for Computational Linguistics. 2021. p. 671–688, https://www.scopus.com/authid/detail.uri?authorId=57028066100.
DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.59.

18. Zhang, J., Lan, Z. & He, J. “Contrastive learning of sentence embeddings from scratch”. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*. *Association for Computational*. *Linguistics Singapore*. 2023. p. 3916–3932, https://www.scopus.com/authid/detail.uri?authorId=58313647000. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.238.

19. Gao, T., Yao, X. & Chen, D. “Simcse: Simple contrastive learning of sentence embeddings”. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Punta Cana, Dominican Republic. Association for Computational Linguistics. 2021. p 6894–6910, https://www.scopus.com/authid/detail.uri?authorId=57211568031.
DOI: https://doi.org/10.18653/v1/2021.emnlp-main.552.

20. Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J. & Weng, L. “Text and code embeddings by contrastive pre-training”. 2022. DOI: https://doi.org/10.48550/arXiv.2201.10005.

21. Giorgi, J., Nitski, O., Wang, B. & Bader, G. “Declutr: Deep contrastive learning for unsupervised textual representations”. *In: proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2020. p. 879–895, https://www.scopus.com/authid/detail.uri?authorId=57204787923.
DOI: https://doi.org/10.18653/v1/2021.acl-long.72.

22. Tiwari, A. & Nalamwar, S, “A review of resume analysis and job description matching using machine learning”. *International Journal on Recent and Innovation Trends in Computing and Communication*. 2024; 12 (2): p. 247–250.

23. Bhatia, V., Rawat, P., Kumar, A. & Shah, R. R. “End-to-End resume parsing and finding candidates for a job description using BERT”. 2019. DOI: https://doi.org/10.48550/arXiv.1910.03089.

24. Luo, Y., Zhang, H., Wen, Y. & Zhang, X. “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*. 2019. p. 1101–1110. DOI: https://doi.org/10.1145/3357384.3357899.

25. Bocharova, M. Y. & Malakhov, E. V. “CapStyleBERT: Incorporating capitalization and style information into BERT for enhanced resumes parsing”. *In Proceedings of the 13th International Conference on So*ftware and Computer Applications. 2024. DOI: https://doi.org/10.1145/3651781.3651820.

26. Abdin, M., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H. & Zhou, X. “Phi-3 technical report: A highly capable language model locally on your phone”. 2024. DOI: https://doi.org/10.48550/arXiv.2404.14219.

27. “ResJobFit end-to-end artificial neural networks based technology for job-resume matching”. – Available from: https://github.com/maiiabocharova/ResJobFit. – [Accessed: Nov, 2024].

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Use of artificial intelligence: The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research method section**.**

Received 11.09.2024
Received after revision 14.11.2024
Accepted 21.11.2025

DOI: https://doi.org/10.15276/aait.07.2024.27
УДК: 004.89

ResJobFit – наскрізна технологія на основі штучних нейронних мереж для підбору вакансій та резюме

Бочарова Майя Юріївна[1)]
ORCID: https://orcid.org/0009-0004-3875-5019; bocharova.maiia@gmail.com. Scopus Author ID: 57193357730
Малахов Євгеній Валерійович[1)]
ORCID: https://orcid.org/0000-0002-9314-6062; eugene.malakhov@onu.edu.ua. Scopus Author ID: 56905389000
[1)] Одеський національний університет імені І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна

АНОТАЦІЯ

Зі зростанням популярності онлайн-рекрутингу все більшого значення набуває якісний підбір кандидатів на вакансії. Через різний досвід, вимоги до освіти та спеціалізації, а також вимоги щодо місцезнаходження, зазначені в оголошенні про вакансію, для якісного зіставлення та ранжування кандидатів необхідно враховувати різні аспекти. Було показано, що до зіставлення резюме та вакансій можна підходити як до проблеми класифікації пар, а також як до пошуку семантичної схожості на основі представлень даних. У той час як класифікаційні підходи обробляють кожну пару вакансія-резюме послідовно, що призводить до квадратичної часової складності, незалежні текстові представлення та ранжування є набагато ефективнішим та масштабованим рішенням, оскільки мають лінійну часову складність. У цій статті використано ранжування за семантичною схожістю для оцінювання кандидатів на відповідність вакансіям. Запропоновано ResJobFit - наскрізну технологію на основі штучних нейронних мереж для зіставлення вакансій та резюме. Технологія ResJobFit складається з моделей сегментації, парсингу, сумаризації та модулю представлення текстів в домені управління персоналом, а також їхніх результатів (вектор та атрибути, що визначають кожне резюме або оголошення про роботу) і векторної бази даних, в якій зберігаються записи. Впроваджено некероване навчання текстових представлень для HR-домену, що інкапсулює дві нові навчальні задачі - внутрішньо- та міжсекційне контрастне вирівнювання. Попередньо навчену BERT-модель адаптовано шляхом навчання її узгоджувати розділи резюме, що містять резюме (summary) частину з останнім місцем роботи, а також частини тієї ж самої вакансії або розділу про роботу. В якості базових моделей були використані TFIDF, BERT, E5 та GTE. Запропоновану стратегію навчання без нагляду порівнювали з підходами SimCSE, DeCLUTR та ConFit. Як метрики для вимірювання точності розробленого алгоритму використано NDCG, MAP та MRR. Показано, що нова мета навчання дозволяє досягти значного покращення порівняно з іншими підходами до навчання без нагляду. Покращення на 11% в NDCG було досягнуто завдяки адаптації стратегії навчання DeCLUTR для HR-домену на основі використання структури резюме порівняно з класичною стратегією навчання DeCLUTR в задачі ранжування згенерованими великою мовною моделлю резюме (summary) вакансій та резюме. 2 % та 6 % покращення було досягнуто в задачі ранжування повнтекстових вакансій та резюме завдяки використанню ResJobFit технології та ResJobFit з узгодженням вимог у порівнянні з найсучаснішою моделлю ConFit.

Ключові слова: інформаційні системи; машинне навчання; обробка природної мови; трансформери; текстові вкладення; інформаційний пошук

ABOUT THE AUTHORS

Maiia Y. Bocharova - Postgraduate, Department of Mathematical Support of Computer Systems. Odesa I. I. Mechnikov National University, 2, Dvoryanska Street. Odesa, 65082, Ukraine
ORCID: https://orcid.org/0009-0004-3875-5019; bocharova.maiia@gmail.com. Scopus Author ID: 57193357730
Research field: Natural Language Processing; similarity search; machine learning; data mining, software engineering

Бочарова Майя Юріївна - аспірант, кафедра Математичного забезпечення комп'ютерних систем. Одеський національний університет імені І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна

Eugene V. Malakhov - Doctor of Sciences (Eng.), Professor, Head of Department of Mathematical Support of Computer Systems. Odesa I. I. Mechnikov National University, 2 Dvoryanska Street. Odesa, 65082, Ukraine
ORCID: https://orcid.org/0000-0002-9314-6062; eugene.malakhov@onu.edu.ua. Scopus Author ID: 56905389000
Research field: Databases theory; metamodeling, the methods of data mining and other data structuring, data processing methods

Малахов Євгеній Валерійович - доктор технічних наук, професор, завідувач кафедри Математичного забезпечення комп'ютерних систем. Одеський національний університет імені І. І. Мечникова, вул. Дворянська, 2. Одеса, 65082, Україна