

Information Retrieval Using Domain Adapted Language Models: Application to Resume Documents for HR Recruitment Assistance

Zakaria Bouhoun(✉), Theo Guerrois, Xianli Li, Mouna Baker, Nassara Elhadji Ille Gado, Emir Roumili, Francesco Vitillo, Lies Benmiloud Bechet, and Robert Plana

Assystem EOS, Courbevoie, France
{zbouhoun,tguerrois,xli,mbaker,nelhadji,eroumili,fvitillo, lbenmiloud,rplana}@assystem.com
https://www.assystem.com

Abstract. To attract and hire the most talented candidate who fit a company's requirements and have the potential to drive its growth, HR professionals spend a significant amount of time sourcing candidates, reviewing their resumes, and assessing their experiences against the needs of open positions. However, with the rise of online recruitment, this process has become increasingly challenging to manage manually. Therefore, job-resume matching has emerged as a crucial technique to automate and streamline the recruitment process. By leveraging machine learning algorithms, job-resume matching allows recruiters to quickly identify the most qualified candidates by comparing their skills and experiences with the requirements of vacant positions. The challenge today is to develop an approach that can effectively encode the semantic meaning of textual information in job descriptions and candidate resumes. In this paper, we propose a novel approach that combines TSDAE and BERT models to create more specialized sentence embeddings for HR data. This approach is powered by clean data extracted using a dedicated CV parsing tool. We also present two methods to automatically create labelled data for training the model. The benchmark of our method against real data shows an improvement of up to 11 points compared to the baseline model without TSDAE, highlighting the effectiveness of our proposed approach.

Keywords: text embedding · job-resume matching · model domain adaptation

1 Introduction

Skills management is a strong strategic stake for all companies. It brings a powerful competitive and economic advantage. In recent years, companies have adopted new recruitment methods to adapt to technological and social changes.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 440–457, 2023.
https://doi.org/10.1007/978-3-031-37105-9_30

Online tools, such as professional social networks and online recruitment platforms, have gradually replaced traditional methods, such as newspaper advertising and recruitment agencies.

Today HR (Human Resources) teams often work on time-consuming tasks such as sourcing on social networks or the empirical construction of skills maps, taking time away from more value-added tasks such as the relational and recruitment strategy. The search for CVs (Curriculum Vitae) also concerns sales representatives who must build up files in response to calls for tender. However, this task must often be carried out in a hurry in view of the required response time. In the digital age era, turning to data science solutions is a necessary approach to improve productivity of human resources management.

In this context, Assystem, an industrial consulting company focused on energy (especially nuclear) and digital transition, developed an application called Emoby which is supported by artificial intelligence. Emoby offers a new approach to human resources management and the possibility of cross-referencing internal and external sources of information to define and anticipate customer's technical needs, better allocate resources, forecast tomorrow's jobs, and identify development opportunities. The tool is based on the analysis of data such as CVs, skills maps and job descriptions. Emoby uses the CV matching which is a technology that allows to find the most relevant profile(s) for an expressed need. This feature uses machine learning techniques to identify the most likely matches by comparing the skills and experiences of candidates, which are in resumes, with the requirements of vacant positions which are in job descriptions. It helps automating and speeding up the recruiting process, reducing the time and resources needed to manually go through hundreds of resumes. The goal is to find the candidates who have the most suitable profile for a given position, to facilitate the selection process for employers.

The Emoby matching algorithm was initially developed using the TF-IDF (Term Frequency-Inverse Document Frequency) method followed by LSI (Latent Semantic Indexing) [8]. TF-IDF is a widely used technique in information retrieval that determines the most important words or phrases in a corpus of documents and ranks documents based on their relevance to a given query. While TF-IDF has many advantages, it suffers from some significant drawbacks, such as sparsity, which makes the matrix computation expensive. Furthermore, it does not incorporate semantic information, which is critical for high-quality matching.

To address the sparsity issue, we employed LSI, which is a dimensionality reduction technique that can group together terms with similar meanings and provide a more compact representation of the data. Additionally, LSI can help to incorporate semantic information into TF-IDF and improve the quality of the matching process. However, the TF-IDF followed by LSI approach still suffers from important problems such as language dependence, vocabulary limitation, and the cost of adding new documents. As an international company, Assystem receives CVs from various countries, for example those written in French, English, and Russian, etc. In this case, TF-IDF creates different representations for each language, even if the words have the same meaning. Additionally, since

Assystem works in various sectors, such as nuclear, defense, building, rail, and others, a large TF-IDF matrix with a high number of columns is required to capture the vocabulary from various industries and languages. This large TF-IDF matrix becomes prohibitively expensive and unsuitable for adding new CVs frequently.

To overcome the limitations of the previous matching algorithm, advanced embedding techniques such as CNN (Convolutional Neural Network) [22], BERT (Bidirectional Encoder Representations from Transformers) [1], and Sentence BERT (SBERT) [11] models have been proposed. These techniques allow for accurate capturing of the semantic and contextual meaning of CVs. Additionally, GNN (graph neural network) approaches [19,21] can be combined with embedding techniques to represent the hiring history and interactions between candidates and recruiters.

However, the implementation of these advanced techniques requires certain considerations:

- A sufficient amount of high-quality labeled data is needed to train the models. Obtaining labeled data can be a challenging and time-consuming process, especially in the case of multilingual and multi-sector recruitment.
- Accurate extraction of text from resumes regardless of the structure, the language, or the format of the resume is crucial for effective matching. This may require preprocessing techniques such as OCR (optical character recognition) and NLP (natural language processing) to ensure that the data is in a suitable format for the models.
- Using more specialized models for the HR domain may be necessary to achieve optimal results. These models would need to be adapted on HR specific data to better capture the nuances of the industry and improve the accuracy of the matching process.

To address the challenges faced by current resume matching algorithms, we propose a novel approach that leverages the SBERT model [15] for generating embeddings. Our approach aims to optimize the use of SBERT by applying a technique called TSDAE (Transformer-based Sequential Denoising Auto-Encoder) [17] to enable the model to adapt to various sectors of candidates and accurately capture the semantic meaning of resumes. In addition, we recognize the importance of high-quality data preparation in achieving better results. Thus, we utilize a specialized resume parser that can accurately extract text from resumes, ensuring that our model is trained on clean and reliable data. To further improve our approach, we also introduce a method for automatically generating additional training data for the model. By doing so, we increase the amount of labeled data available for training, which can be particularly valuable in scenarios where obtaining labeled data is a challenging and time-consuming process. Our proposed approach shows promising results in overcoming the limitations of traditional resume matching algorithms and improving the accuracy of candidate selection in recruitment processes. The architecture of our proposed approach is illustrated in Fig. 1 below.

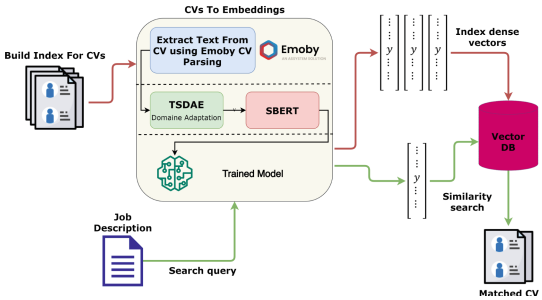

Fig. 1. The overall architecture of our proposed approach

2 Related Works

The task of matching resumes and job descriptions has been the subject of extensive research in recent years. With the increasing number of job postings and job seekers, it has become crucial to find the most suitable candidate for each job offer quickly and efficiently. Various approaches have been proposed to address this challenge, with the common goal of utilizing effective encoding methods to represent the information contained in resumes and job descriptions in a structured and meaningful way, enabling efficient and accurate comparison.

In this section, we will underline the related literature in this research area and discuss the different techniques and algorithms that have been proposed to assist HR in the hiring process. A well-known approach to represent textual data is TF-IDF. [16] suggest clustering resumes into different categories corresponding to different domains of expertise, and then uses a TF-IDF method to represent the text extracted in each resume. The TF-IDF of a resume is then used for two different tasks: binary classification of the resume (accepted, rejected) using models such as Random Forest [2], and ranking using the cosine similarity score. In [13], the same approach based on TF-IDF is used in combination with other features extracted from the resume and JD (job description) including words count, skills, years of experience, spelling errors, and document layout feature. Based on these features, two models are built, one to perform a binary classification (accepted/rejected) and the second for CV ranking using a weighted score function across all features. The approach proposed by [18] is also based on TF-IDF for text representation but applied separately to each section part, including experience, core skills, skill keywords and activity keywords, which means that a CV will not be represented by a single TF-IDF matrix, but by several TF-IDF matrices. To calculate the final ranking, a matching score between each section and the job description is first calculated and then the average of these scores is considered as the final score. To help users find more precise results, the approach in [18] go further and used a Knowledge Graph constructed from relevant information extracted from CV and JD using a NER (Named-entity recognition) based on a fine-tuned BERT model [3] on HR data using the MLM (Masked Language Model) task. In the same objective of building an efficient representation

of the CV and JD, [22] proposes an approach to create text embedding using CNN models. The system consists of two stages: Candidate Retrieval that uses a fused embedding strategy to learn representations from CV texts, job skill information graphs and geolocation for candidates and JDs. For this purpose, a CNN model fed by a domain-specific vocabulary for human resources provided by the word2vec (word to vector) model [5] is used to train the embeddings. The second stage, named Reranking, uses a FAISS (Facebook AI Similarity Search) [10] index for clustering and compressing the embeddings, which also allows searching for the closest candidates in real time.

A new approach is proposed in [21] based on GNN. The algorithm consists of constructing a dual-perspective interaction graph that incorporates two different nodes for each candidate (or job): one captures a candidate's preferences for selecting a job and the other captures an employer's preferences for selecting a candidate. Thus, this dual perspective graph approach allows for the modelling of two-way selection intentions. In [19], the authors propose a new method that uses co-attention neural networks in conjunction with GNNs to create semantic representations of candidate profiles and recruitment history. To achieve full semantic representations of both resumes and job postings, they utilize mashRNN (Multi-depth Attention-based Hierarchical Recurrent Neural Network) [9], which extracts data from various structural levels of the documents. The co-attention neural networks are then used to assess the semantic similarity between resumes and job postings. Additionally, GNNs are utilized to generate representations of previous hiring experiences and integrate them into the matching calculation between candidates and job vacancies. Another approach that combined neural networks and GNN is proposed by [1]. The authors used a text matching model based on a pre-trained BERT to obtain a textual representation of resumes and JDs. They also built a relation-based matching model that uses a GNN to learn node representations based on a JD-CV historical matching. The proposed approach also uses a co-teaching mechanism to reduce noise in the training data and improve the model's capability to learn from limited or sparse interaction data.

In [11], the authors presented a new approach of resume matching based on SBERT (Sentence BERT) model. The SBERT has been established as a state-of-the-art model for producing high-quality embeddings that perform extremely well on various natural language processing tasks, including semantic textual similarity and re-ranking tasks [14]. The authors in [11] fine-tuned a multilingual BERT model using the Siamese networks on labelled data extracted from their HR solution. The authors also performed a comparative analysis of the embeddings generated by the TF-IDF, BERT and the fine-tuned SBERT model, using identical data from the HR solution, and the results demonstrated a better performance of the fine-tuned SBERT approach.

3 Contributions

This section highlights the main contributions made during the research work conducted. These contributions include text extraction using a resume parser,

the creation of datasets, and the adaptation and training of the model domain. These efforts were essential to the success of the project and are detailed below.

3.1 Text Extraction Using a Resume Parser

To produce meaningful insights and accurate predictions in data science projects, it is essential to prepare and produce a high-quality dataset. A good dataset should be representative of the target population, complete, accurate, and unbiased. Preparing such a dataset is often a complex and challenging process, requiring data scientists to employ various techniques to preprocess and standardize the data.

In our case, the resumes are stored in various file formats, including PDF, DOCX, and image files. This diversity in file types can complicate the process of extracting and analyzing the data, as each format requires a different approach for data extraction. Additionally, each candidate's CV is unique in terms of the domain they have worked in, the language they use, and the overall structure of their document. Therefore, before any analysis can be performed on the CVs, a pre-processing step is required to standardize the data and ensure that it is ready for analysis. This process may include steps such as layout analysis, text recognition and extraction, and entity recognition.

The first stage of the proposed solution is based on the DiT (Document Image Transformer) model [12] which is the first model trained on a large amount of unlabeled text documents. This approach is crucial in the field of document AI due to the time-consuming task of annotating documents and the limited availability of annotated data compared to the wide variety of documents such as CVs, invoices, and articles. The DiT model is considered as document model (similar to language models in the NLP domain) ready to be used as a backbone to feed other models that can perform various tasks, such as document classification or layout analysis. In this project, we used the DiT model as a backbone to extract informative features from CVs, which were then utilized to fine-tune Mask R-CNN (Mask Region-Based Convolutional Neural Networks) [6] as a detection framework on annotated CV data to perform segmentation tasks.

After detecting the sections in the CV, the task of assigning labels to each section is performed through a classification process. Given the visually rich nature of CVs with different layouts and text fonts, a multi-model approach is adopted to leverage both linguistic and spatial features. The commonly used LayoutLM (Layout Language Model) [20] model, inspired by BERT architecture, is ideal for this purpose. LayoutLM incorporates spatial information and visual embeddings to enhance text classification and information extraction tasks. Furthermore, LayoutLM is trained on a significant amount of unlabeled raw documents using a self-supervised approach that enables the model to learn language and layout representations without requiring annotated data. In the second stage for the Emoby CV Parsing, we fine-tune the LayoutLM model on the task of section classification, taking advantage of the text features and spatial features to improve classification accuracy.

We evaluated the performance of the first stage of our system on a test set comprising CVs with varying structures from easy to complex, and it showed a score of 86% of MAP (Mean Average Precision). For the second stage related to section classification, we achieved an F1-Score of 90% for the most important sections, such as Experience, Professional summary, and Personal Information. Additionally, we achieved F1-scores of 85% for Education and 80% for Skills. These promising results encourage us to further develop our approach by incorporating more data, refining the system architecture, and expanding the scope of the project to improve system performance.

3.2 Dataset Creation

According to the official Huggingface explanation [4], to train a sentence transformer model, it is essential to understand how to introduce data into the model and prepare the dataset according to our needs. Furthermore, each configuration has a loss function related to the construction of the dataset. Indeed, there are four general forms to describe different database structures. These also differ according to the purpose and use of artificial intelligence.

Thus, we find:

1. Pairs of sentences with a similarity score for each, from 0 for a contradiction to 2 for an implication.
2. A dataset of similar sentences pairs without similarity score.
3. A dataset of sentences having a discrete score indicating the class to which the sentence belongs.
4. A dataset constructed as triplets (anchor sentence, similar sentence, dissimilar sentence) with no classification nor score.

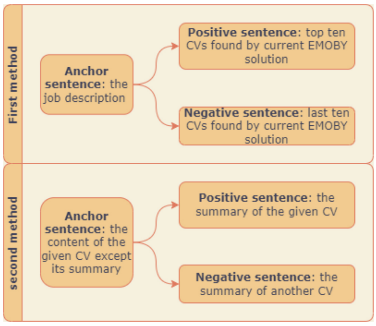

Fig. 2. Dataset structure of our two methods

The present study presents a visual representation of the final dataset structures for each method implemented, as illustrated in Fig. 2. The initial approach

involves utilizing the structure proposed by Huggingface, where the job description serves as the anchor sentence, the top ten CVs identified by the current Emoby solution serve as positive sentences, and the last ten CVs identified serve as negative sentences. The second approach entails generating analogous pairs of sentences comprising the contents of a provided CV, except for the summary section and the actual summary. The summary of an alternative CV serves as the negative sentence in this method. Figure 2 demonstrates in a clearer way the final dataset structure of each implemented method in this paper.

During training, positive and negative samples are commonly utilized to enhance the model's ability to comprehend the similarity between different sentences. Positive samples are sentences that are similar to each other, whereas negative samples are dissimilar sentences. Hard negative samples are samples that are challenging for the model to classify correctly, resulting in a significant loss during the training process. Consequently, they are particularly beneficial for accelerating convergence and improving the model's ability to learn the decision boundary. In information retrieval tasks, positive samples are predefined, whereas negative samples can be extracted more easily.

One approach to improve the performance of an existing solution (Emoby) is to use TF-IDF+LSI to retrieve relevant resumes, where the top ten retrieved resumes are considered positive samples, and the last ten resumes in the rank are regarded as hard negative samples. Such dataset structure matches the last case above: we form a dataset of triplets (anchor sentence, similar sentence, dissimilar sentence) with no classification nor score.

In scenarios where similar sentence pairs exist, the summary of a particular resume is utilized as a positive sentence, while negative samples are randomly chosen from the summaries of other resumes. These data structures correspond to various cases of triplets or pairs without classification or similarity score.

The creation of a dataset faces several challenges to overcome or at least to consider. One of the primary challenges is understanding and identifying the relevant information from resumes, which can vary in organization and structure depending on who is writing them. In fact, capturing the correct information, therefore, becomes complicated. To address this challenge, we are using the CV parsing solution describe in Sect. 3.1 that meets our needs.

Another challenge is in the lexical field used. This is specific to the nuclear energy sector. That means some terms may be redundant, so it is necessary to understand the CVs correctly, in order to be sure that the contexts are well affiliated with the right situations. Another challenge is the variation in language structures and grammatical rules across languages. For example, some languages have more complex grammar structures than others, or use different scripts or writing systems.

Moreover, it is important to ensure that there is no gender or language favoritism. This challenge is very important as this could lead to unfair representation of candidates. As French words are involved, adjectives and other words should not influence models. Parity is created to avoid any favoritism.

To represent candidates fairly, the model needs to learn with data from both genders.

One of the key challenges in matching resumes and job descriptions is dealing with multilingual data. Models must be able to handle this variety of languages. To better take this into account, multilingual models are used and amounts of high-quality data in each language are collected. The Assystem job search website includes resumes in French, English, Arabic, etc. The fact that we only use data from our sites is related to the next challenge.

The GDPR (General Data Protection Regulation) is a set of regulations that aim to protect the personal data of individuals in the EU (European Union). As they contain personal data, the consent from the individual is needed. To ensure the security of data, they are collected and analyzed with internally made solutions.

Dataset Based on the Emoby Solution. The first method involves 800 sets of [queries, positives, negatives], in which job descriptions (queries) are paired with information from resumes (positives, negatives). In fact, CVs that are identified as "positive" i.e., they correspond to desired characteristics involved in the field of work, are used as sentences and contexts that match job descriptions. The ones chosen as "negative" i.e., they do not apply to the job, are used to fine-tuned and exclude fields that are not linked to job descriptions. The following format is used:

'query': description of the job, 'positive': List[resume], 'negative': List[resume]

In [4], it is advisable to use TripletLoss for this type of architecture. This is a popular loss function used in tasks that involve learning a similarity metric between triplets of inputs. This loss function encourages the model to learn embeddings that group together similar examples and separate dissimilar examples. The TripletLoss function tries to minimize the distance between the anchor and positive examples while maximizing the distance between the anchor and negative examples.

To train different implemented models, a dataset needs to be built. CVs and vacancies must be collected automatically and quickly.

To achieve this, Emoby, a deployed instance with more than 50 000 active users, is used. This functional version of Emoby is based on the TF-IDF+LSI algorithm which, in addition to matching, will calculate a percentage of correspondence between the CVs found and the job descriptions entered by an HR user. This score will be used to be sure that CV match more than the threshold fixed. The intensive use of the algorithm that sits within the software, i.e., the processes of experts in the field of HR, provides us with solid elements on which to base our increase in data of interest for our training.

Our Human Resources provide job description to be sure that the query suits our nuclear field of work. A pipeline is created to collect resume. It will automatically select a JD, remove special character in the text, do the research in the Emoby database, collect the required CVs, and put them in the right field.

To see if the data collected are relevant. Once all CVs and their matching scores have been retrieved, Fig. 3 emerge from these data. graphics have been made. They represent the distribution of the positive and negative matching scores. To improve the visibility of graphics, each color represents a score of matching. The percentage of scores obtained after data collection is marked for each color. For example, the green color in the Fig. 3 indicate that 14.5% of CV match a JD with a score of 84.

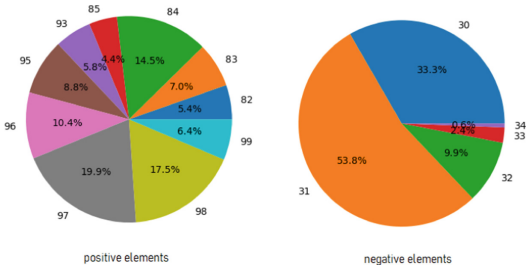

Fig. 3. Distribution of the matching score for all positive and negative items

A confidence interval appears and allows to view the distribution of the data obtained with Emoby. Positive scores are all above 81% of matching i.e., the similarity scores generated for the pairwise JD/resume is high, so similar words are found in both documents. Negative ones do not exceed 34% of matching i.e., the lexicon of the documents is quite disparate

Dataset Based on the CV and Its Summary. Different from the first method based on the correspondence between a job description and a CV, for the second method, we would like to build a dataset taking advantage of the very nature of the CV. We noticed that most resumes contain a summary resuming the experience and education of the given candidate, which reforms a candidate's most valuable qualities differently and accurately. This very nature of a CV summary makes it a positive sentence to the given resume. Thus, during training, our model will consider one's CV and the summary as two sentences having the same meaning. The output model could focus better on the resemblance between a job description and a CV. Since we can't know the similarity score between the CV and its summary, the second configuration, a dataset of similar sentences pairs, is the most suitable option.

It is recommended to use MultipleNegativesRankingLoss [7], one of the most recommended loss functions for datasets consisting of positive pairs. In the case of the second method, the content of a given CV except its summary and the summary form a pair. This loss function takes sentence pairs as input, while summaries related to other summaries are considered as negative samples. It is also possible to determine one or multiple hard negative samples per pair to improve the model's performance.

In contrast to the first method, the second method is independent of the current Emoby solution and involves the acquisition of CVs from the Assystem job search website to supplement the training dataset. However, this approach presents certain challenges as the dataset includes a mixture of cover letters, references, job descriptions, and resumes. Due to the presence of extraneous data, it is imperative to conduct pre-processing of the CV to eliminate irrelevant information before the training process.

To retrieve and classify different sections present in a resume, we use CV parsing, the tool presented in Sect. 3.1. Based on its output, we have identified different types of content, such as education, experience, skills, and personal information etc.

First, we should identify the main characteristics of a resume and how they differ from other types of documents, such as a cover letter and certification. After observing the CV parsing results, we found several missing parts in non-resume documents, especially "education" or "experience", which are the two most valuable content in resumes. The absence of these two allows us not only to remove most cover letters and certifications, but also to filter out resumes that lack these essential elements of education. Also, if the summary is less than 30 words, then it does not contain enough information for training, and we will not consider this CV.

Also, this solution does not preserve a specific tree structure compared to the previous one. Thanks to CV parsing, we managed to identify different content in the CV. We also ranked them based on their importance in recruiting. The reason behind this action is that BERT-based Transformers [3] have tokens limitation, which means a BERT-based model can only take a max length limit of 512 tokens, in case having a long CV and important information gets cut out, we put the important ones like education and experience in the beginning.

Each category is assigned with a weight and priority. So here is the order of priority: firstly, the personal information of the candidates containing their address ranks top to match the most suitable candidates in mobility. Then their project and experience. Their skills, language, and education/certifications rank third. Additionally, this filtering for empty or missing categories can clean up databases of certificates, cover letters, or job descriptions.

In the end, the constructed dataset contains around 9000 CVs, which is useful and usable information for the training of TSDAE and SBERT model.

3.3 Model Domain Adaptation

In this paper, our primary method for matching job descriptions and CVs is using TSDAE as a pre-trained algorithm to improve SBERT models, which has shown superior performance for domain adaptation compared to classical pre-training tasks such as Mask Language Model.

During the pretraining, TSDAE damages the input data by randomly deleting around 60% of tokens as the input noise. The encoder generates fixed-sized vectors, and the decoder tries to reconstruct the original sentences. During the process, the TSDAE model captured well the semantics of input data.

One notable advantage of TSDAE is its ability to function without the need for enormous amounts of labelled data, which are often scarce and costly. Being an unsupervised method, it can discover hidden patterns or groups within the data that may not be immediately apparent. Additionally, these algorithms can adapt to new data or different application domains without requiring a complete retraining of the model, which is particularly important for our constantly evolving CV database.

Furthermore, the introduction of noise with this method can aid in better understanding the CVs. By training the model to reconstruct a corpus, it can account for typographical errors, leading to a better comprehension of the context and situations contained within the data.

As per the TSDAE paper, fine-tuned models are deemed inadequate for TSDAE implementation due to their pre-adaptation to specific sentence embedding tasks. Instead, the recommendation is to use a pre-trained language model, notably the BERT-base-uncased model, which has been proven to achieve superior results. To cater to the linguistic diversity of our CV dataset, which includes English, French, and Arabic, we employed the BERT-base-multilingual-uncased model in the first method. Given the significantly larger size of the dataset in the second method, the use of a BERT-base-uncased model, which is too large to achieve optimal performance, was not viable. Instead, we opted for the second-best pre-trained model, namely the Distil-BERT-base-multilingual-cased model, to effectuate the second method.

As previously mentioned, the SBERT model was fine-tuned using TripletLoss in the first method, while MultipleNegativesRankingLoss was used in the second method. However, it is worthwhile to investigate whether TripletLoss can yield superior results in the second method and whether MultipleNegativesRankingLoss can be more effective in the first method. To evaluate the performance of the two pre-trained models under different training loss functions, we fine-tuned them with both MultipleNegativesRankingLoss and TripletLoss based on TSDAE. In addition, to demonstrate the effectiveness of TSDAE, we also fine-tuned two models without utilizing TSDAE.

After completing the stage of fine-tuning, it is imperative to properly evaluate our model through the application of appropriate metrics and methods. A plethora of evaluators for various natural language processing tasks are provided by the sentence transformers package.

When employing TripletLoss in the fine-tuning process, which optimizes the distance between embeddings in the vectorial space, it is advisable to utilize the TripletEvaluator. This evaluator calculates the distance between the embedding of the anchor sentence and those of both the positive and negative sentences. The resulting accuracy is determined by computing the percentage of instances where the distance between the anchor and positive sentence is smaller than the distance between the anchor and negative sentence.

We utilize the RerankingEvaluator for the MultipleNegativesRankingLoss, which optimizes using similarity functions like cosine similarity. This evaluator reorders the mixed list of positive and negative sentences based on the cosine

similarity score. The metrics used for evaluation are MAP and MRR@10 (Mean Reciprocal Rank).

MAP considers the number of correct predictions and assigns varying degrees of importance to these predictions. To achieve a higher score, a greater number of correct predictions are required, with greater emphasis placed on the accuracy of the most important predictions. For instance, if we assume that the right prediction is denoted by 1 and a wrong prediction is denoted by 0, then the sequence [1, 1, 1, 0, 0] would receive a superior score of MAP to [1, 0, 1, 0, 1], despite both having an identical number of good predictions.

MRR@10, on the other hand, estimates the likelihood of the first correct prediction being ranked within the number specified after the "@" symbol, in this case, the top ten positions. In this study, only MRR@10 was utilized to evaluate the performance of the models. Therefore, in the subsequent sections of this paper, the term MRR will specifically denote MRR@10.

Results for the First Method. To display results, both loss functions are modified to include wandb (Weights and Biases[1]). This is a platform for developers to track experiments and visualize results. Thanks to it, hyperparameters are easily tracked at each epoch, and the accuracy of the model can be recorded. This enables the model to be adapted by adjusting the parameters in response to the captured accuracy data.

Firstly, a dataset to train the TSDAE model needs to be built in line with the first method. Two datasets are constructed, an unlabeled one to train the model and the other, built by Human Resources, is a "golden" dataset to test the model.

Table 1. Metrics of model MRR and MAP on validation data (200 elements).

Model Name	TSDAE	loss function	MAP	MRR
BERT-base-multilingual	No	MultipleNegativesRankingLoss	0.85	0.91
BERT-base-multilingual	No	TripletLoss	0.89	0.92
BERT-base-multilingual	**Yes**	**MultipleNegativesRankingLoss**	**0.93**	**0.98**
BERT-base-multilingual	Yes	TripletLoss	0.90	0.97

To monitor and measure the performance of models in a reranking task, the benchmark [23] used two metrics, MAP and MRR, which are usually employed to evaluate performance of the models on their evaluation dataset and the golden dataset.

As a reminder, all four models are trained with the same dataset. Overall, incorporating a TSDAE step resulted in an increase in accuracy, despite the limited size of the database, which consisted of only 800 unlabeled samples.

[1] https://wandb.ai/site.

However, this small dataset may have a double-edged effect on the outcomes. On the one hand, the addition of a TSDAE layer significantly improved the quality of the embeddings, but on the other hand, the insufficient number of samples could lead to overfitting of the model due to a certain lack of diversity. The database should represent most of the possible cases, so the database created by HR and the validation dataset will show if the model has truly understood the data and if it is applicable on real data.

Table 1 presents the results of the metric on the validation set, which indicate that the MAP (mean average precision) value is greater for the model incorporating TSDAE in comparison to the one without the addition of the layer. Specifically, the MAP value improved from 0.89 for the model without TSDAE to 0.93 for the model with TSDAE. This finding indicates that the model incorporating TSDAE can identify a higher number of relevant CVs for each job description in comparison to the model without TSDAE. Additionally, the MRR value is also higher for the model with TSDAE on the validation set, increasing from 0.92 to 0.98. In the context of reranking tasks, MRR is the average of the reciprocal of the rank of the first relevant result across all queries. Therefore, the improved MRR value for the model with TSDAE indicates that relevant CVs are also graded better since they appear higher up in the list of results.

Using the same evaluation metric, the Human Resources provided dataset was utilized to evaluate the performance of the models. This dataset consisted of manually selected candidates that matched or did not correspond to the job descriptions, representing real cases. However, the dataset only contained around 40 items, which limits our ability to identify trends or make definitive assertions regarding the model's performance. While the quality of the dataset's construction may provide insight into the models' real-world performance, the small sample size does not necessarily enable us to draw conclusions or detect trends that the models may capture.

Table 2. Metrics MAP and MRR on 'golden' dataset.

Model Name	TSDAE	loss function	MAP	MRR
BERT-base-multilingual	No	MultipleNegativesRankingLoss	0.83	0.81
BERT-base-multilingual	No	TripletLoss	0.86	0.88
BERT-base-multilingual	**Yes**	**MultipleNegativesRankingLoss**	**0.89**	**0.91**
BERT-base-multilingual	Yes	TripletLoss	0.85	0.91

As we can see in Table 2, the TSDAE pre-trained model with MultipleNegativesRankingLoss has a better outcome on data that the model had never seen before, i.e., the "golden" dataset. MAP and MRR increased by 0.03 and 0.03 respectively compared to the model without the introduction of a TSDAE model.

In both distinct datasets, the incorporation of TSDAE in the model architecture resulted in a significant improvement in performance. This improvement

can be attributed to the model's ability to more accurately capture important features and relationships between words or sentences. This enhanced capability of the model is particularly valuable since the database used in the study was created by HR professionals and represents real cases of individuals who have undergone a recruitment process.

Result for the Second Method. The second method of the study implements two different training loss functions to compare their respective performances. Specifically, the study evaluates the efficacy of MultipleNegativesRankingLoss and TripletLoss in training a model. To evaluate the effectiveness of each loss function in training the model, the study uses two different evaluators. The RerankingEvaluator is utilized to evaluate the ranking quality of the documents retrieved when MultipleNegativesRankingLoss is employed. On the other hand, TripletEvaluator is employed to evaluate the quality of the learned embeddings when TripletLoss is employed.

Meanwhile, we would like to know how effective the TSDAE model is, a comparison of two training results with and without the TSDAE model is shown in Table 3 and 4. We trained two models with the same dataset containing 9000 CVs. We observed that the values of both metrics of the training with TSDAE are higher than the one without TSDAE.

Table 3. Comparison of two MultipleNegativesRankingLoss training results evaluated on validation dataset with and without TSDAE, using RerankingEvaluator.

Model Name	TSDAE	loss function	MAP	MRR
DistilBERT-base-multilingual	No	MultipleNegativesRankingLoss	0.74	0.74
DistilBERT-base-multilingual	**Yes**	**MultipleNegativesRankingLoss**	**0.79**	**0.78**

Table 4. Comparison of two TripletLoss training results evaluated on validation dataset with and without TSDAE, using TripletEvaluator.

Model Name	TSDAE	loss function	Mean accuracy
DistilBERT-base-multilingual	No	TripletLoss	0.91
DistilBERT-base-multilingual	**Yes**	**TripletLoss**	**0.92**

It is evident that utilizing TSDAE as a pre-training method yields improved performance of the model upon completion of the training process. The results presented in both tables pertain to the evaluation of the model's performance on the validation dataset at the conclusion of the training phase. However, it is crucial to ascertain the model's efficacy on the previously mentioned 'golden' dataset. The outcomes of such evaluations are as follows (Table 5):

Table 5. Metrics MAP and MRR on 'golden' dataset.

Model Name	TSDAE	loss function	MAP	MRR
DistilBERT-base-multilingual	No	MultipleNegativesRankingLoss	0.75	0.83
DistilBERT-base-multilingual	No	TripletLoss	0.81	0.90
DistilBERT-base-multilingual	**Yes**	**MultipleNegativesRankingLoss**	**0.86**	**0.95**
DistilBERT-base-multilingual	Yes	TripletLoss	0.84	0.90

The present study investigates the impact of pre-training method, TSDAE, on the performance of fine-tuned models using TripletLoss and MultipleNegativesRank-ingLoss functions. Results indicate that incorporating TSDAE as pre-training method leads to a significant improvement in model performance, as evidenced by a 3-point increase in the case of TripletLoss and an 11-point increase in the case of Multi-pleNegativesRankingLoss. This finding underscores the efficacy of TSDAE in the current context. Notably, models trained with MultipleNegativesRankingLoss outperformed those trained with TripletLoss by 2 and 6 points, contingent on the utilization of TSDAE. In sum, the study concludes that TSDAE as pre-training method and MultipleNegatives-RankingLoss as training loss function led to superior model performance.

4 Conclusion

In conclusion, the introduction of TSDAE in the architecture has proven to be an asset for improving the accuracy of predictions on two distinct datasets. By allowing the model to better capture important features and relationships between words or sentences, TSDAE has demonstrated its ability to enhance the performance of natural language processing models. However, the degree of improvement observed in this study was found to be limited in comparison to the outcomes reported in the TSDAE publication. There are two main reasons for this discrepancy. Firstly, the implementation of the second method was hampered by limitations in GPU capacity, precluding the use of a BERT-based model with a large batch size that could have yielded superior results. Instead, a smaller and quicker alternative, the DistilBERT-based-multilingual-cased model, was employed. To address this issue, a more advanced GPU with higher computational capacity would be required.

Secondly, it is essential to consider the number of input elements during both the TSDAE training and fine-tuning phases. For example, for the first method, increasing the number of input elements can lead to a better accuracy, but it also brings new challenges. Moreover, all of this must be in line with the challenges found. This is particularly significant in the context of recruitment processes, where the ability to accurately predict outcomes can have a significant impact on the selection of candidates. The fact that one of the datasets used in this study was created by HR professionals and represents real cases further highlights the practical value of these findings. By incorporating TSDAE into

their models, organizations can improve their recruitment processes and make better-informed decisions when selecting candidates for open positions.

References

1. Bian, S., et al.: 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, pp. 65–74 (2020)
2. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Espejel, O.: Train and fine-tune sentence transformers models (1999). https://huggingface.co/blog/how-to-train-sentence-transformers
5. Goldberg, Y., Levy, O.: Word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
7. Henderson, M., et al.: Efficient natural language response suggestion for smart reply. arXiv preprint arXiv:1705.00652 (2017)
8. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999). https://doi.org/10.1145/312624.312649
9. Jiang, J.Y., Zhang, M., Li, C., Bendersky, M., Golbandi, N., Najork, M.: Semantic text matching for long-form documents. In: The World Wide Web Conference, pp. 795–806 (2019). https://doi.org/10.1145/3308558.3313707
10. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2019). https://doi.org/10.1109/TBDATA.2019.2921572
11. Lavi, D., Medentsiy, V., Graus, D.: conSultantBERT: fine-tuned Siamese sentence-BERT for matching jobs and job seekers. arXiv preprint arXiv:2109.06501 (2021)
12. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: DiT: self-supervised pre-training for document image transformer. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 3530–3539 (2022). https://doi.org/10.1145/3503161.3547911
13. Menacer, M.A., Hamda, F.B., Mighri, G., Hamidene, S.B., Cariou, M.: An interpretable person-job fitting approach based on classification and ranking. In: Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021), pp. 130–138 (2021). https://aclanthology.org/2021.icnlsp-1.15
14. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: MTEB: massive text embedding benchmark. arXiv preprint arXiv:2210.07316 (2022)
15. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). https://arxiv.org/abs/1908.10084
16. Roy, P.K., Chowdhary, S.S., Bhatia, R.: A machine learning approach for automation of resume recommendation system. Procedia Comput. Sci. **167**, 2318–2327 (2020). https://doi.org/10.1016/j.procs.2020.03.284

17. Wang, K., Reimers, N., Gurevych, I.: TSDAE: using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. arXiv preprint arXiv:2104.06979 (2021)
18. Wang, Y., Allouache, Y., Joubert, C.: Analysing CV corpus for finding suitable candidates using knowledge graph and BERT. In: DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications (2021)
19. Wang, Z., Wei, W., Xu, C., Xu, J., Mao, X.L.: Person-job fit estimation from candidate profile and related recruitment history with co-attention neural networks. Neurocomputing **501**, 14–24 (2022). https://doi.org/10.1016/j.neucom.2022.06.012
20. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1192–1200 (2020). https://doi.org/10.1145/3394486.3403172
21. Yang, C., Hou, Y., Song, Y., Zhang, T., Wen, J.R., Zhao, W.X.: Modeling two-way selection preference for person-job fit. In: Proceedings of the 16th ACM Conference on Recommender Systems, pp. 102–112 (2022). https://doi.org/10.1145/3523227.3546752
22. Zhao, J., et al.: Embedding-based recommender system for job to candidate matching on scale. arXiv preprint arXiv:2107.00221 (2021)
23. Zoupanos, S., Kolovos, S., Kanavos, A., Papadimitriou, O., Maragoudakis, M.: Efficient comparison of sentence embeddings. In: Proceedings of the 12th Hellenic Conference on Artificial Intelligence, pp. 1–6 (2022). https://doi.org/10.1145/3549737.3549752