ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MASTER'S THESIS

Skill Understanding: from Self-Entered Profiles to Quantitative Comparison

Author:
SENA NECLA ÇETIN

Academic Supervisor:
PROF. TANJA KÄSER

Company Supervisor:
EMMA LEJAL GLAUDE

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science
in the
School of Computer and Communication Sciences

March 18, 2022

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Abstract

School of Computer and Communication Sciences
Machine Learning for Education Laboratory
Swisscom Digital Lab

Master of Science

Skill Understanding: from Self-Entered Profiles to Quantitative Comparison

by SENA NECLA ÇETIN

In recent years, using machine learning solutions in human resources has gained much interest to help with the various challenging tasks in the field. In this thesis, we introduce a Sentence-BERT (SBERT) model that can compute efficient candidate-job role similarities and consequently find the best-fit candidate for a role. To do so, we construct contextual embeddings of employee skill profiles and job descriptions, and compute cosine similarity between employee-job role pairs. By fine-tuning a BERT model on a job cluster classification task, we overcome the challenges of using a self-entered skill profiles dataset, which contain noise, multilinguality, synonymity, granularity, and incompleteness. We show how our SBERT model performs using 2-D profile embedding plots, and examples of employee-employee and employee-job role pairs along with their similarity scores.

Contents

List of Figures

List of Tables

List of Abbreviations

BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
SBERT	**S**entence-**BERT**
NLP	**N**atural **L**anguage **P**rocessing
BoW	**B**ag-**o**f-**W**ords
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
ML	**M**achine **L**earning
HR	**H**uman **R**esources
t-SNE	**t-**Distributed **S**tochastic **N**eighbor **E**mbedding
LR	**L**ogistic **R**egression
CV	**C**urriculum **V**itae

Dedicated to my dear family…

Chapter 1

Introduction

Using machine learning (ML) to solve complex problems in various business areas has gained much interest over the past few years. Specifically, in Human Resources (HR), some of these problems include finding the best-fit candidate for a role, best-fit team for a project, and recommending trainings to employees. According to the World Economic Forum, 50% of the global workforce will require reskilling by 2025 due to the fast-evolving tecnhological requirements in businesses (Zahidi et al., 2020).

Like many other companies, Swisscom is looking to utilize predictive analytics and data-driven insights for their HR-related tasks. A company tool is used for employees to enter their skills. However, since these skills are entered manually by the employees, they represent a multitude of challenges. First, the skills are entered in four languages, i.e., English, German, French, and Italian. Moreover, different wordings are used to represent similar skills across profiles, e.g., *teamwork* and *collaboration*, and at different levels of granularity, e.g., *machine learning* and *pandas*. Finally, the employee skill profiles are often incomplete, where only a few skills are entered per profile.

Transformer-based deep learning models have become state-of-the-art for many fields since they were introduced in 2017 (Vaswani et al., 2017). Specifically, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), using only the encoder of the Transformer architecture, is nowadays the standard for solving a number of natural language processing (NLP) tasks, including natural language inference, word sense disambiguation, and sentiment classification.

This thesis presents a Sentence-BERT (Reimers and Gurevych, 2019) model that can compute candidate-job role similarity and find the best-fit candidate

for a role using cosine similarity. The pre-trained BERT model is first fine-tuned on classifying employee profiles into job clusters. Then, this fine-tuned model is used to obtain the Sentence-BERT embeddings of the employee profiles using mean pooling. These embeddings are used to compute employee-employee and employee-job role similarities. The model shows promising results that may allow the use cases to be extended, such as finding the best-fit team for a role, and recommending trainings to employees. Since this is the first project utilizing machine learning for HR tasks at Swisscom, our aim is to explore the representaton capability of self-entered employee profiles.

This project has been conducted in collaboration with another research team from Roche. The collaboration aims to investigate and experiment with two different models for employee skills representation. Due to working with text data, the Swisscom team has investigated the use of word embeddings for skills representation. On the other hand, since skills have an inherent hierarchical structure, our collaborators at Roche have investigated the use of knowledge graphs.

In the following, we first describe the background of the NLP methods that we will use and present the related work in the current literature. Then, we introduce the dataset and explain the preprocessing steps. We explain the methods we introduced to classify employees into job clusters and to compute employe-employee and employee-job role similarities, followed by their experimental results on both Swisscom data and external data. We conclude by discussing the results, suggesting potential improvements for future work, and sharing our contributions.

Chapter 2

Background

In this chapter, we define and summarize the background knowledge used throughout this report.

2.1 BERT

BERT (Devlin et al., 2018) is a state-of-the-art natural language representation model that is designed to pretrain representations from an unlabeled large corpus by conditioning on both left and right context in all layers. Using an additional output layer, the model can be finetuned for a variety of tasks such as sequence classification or question answering.

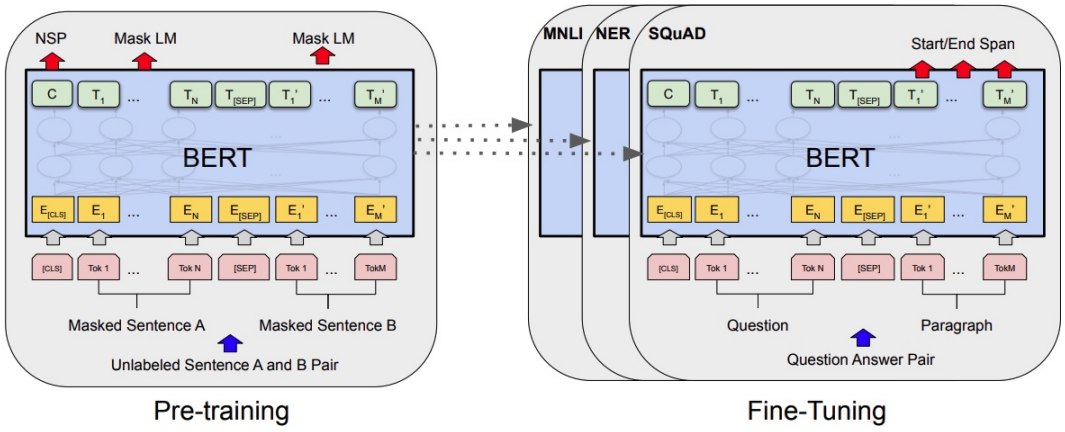

FIGURE 2.1: Figure taken from Devlin et al., 2018. Pre-training and fine-tuning structures for BERT. The same architectures are used in both pre-training and fine-tuning except the output layers. The pre-trained model parameters are fine-tuned for different down-stream tasks.

The BERT model uses a multi-layer bidirectional Transformer architecture based on the original implementation (Vaswani et al., 2017). The attention mechanism used in the Transformer architecture gives BERT a high capacity

to understand context and ambiguity in language by processing any given word in relation to its surrounding words. This is opposed to the traditional word embeddings like GloVe (Pennington, Socher, and Manning, 2014) and Word2Vec (Mikolov et al., 2013), which use a single numeric representation of a word, disregarding context and homonymity. Moreover, attention allows for significantly more parallelization than previous state-of-the-art architectures, including recurrent neural networks (RNNs). This enables training on larger language datasets and contributes to BERT outperforming baseline models on a wide range of natural language understanding tasks.

2.2 Sentence-BERT

Although BERT has become the state-of-the-art method for many natural language processing tasks, since it requires that sentence pairs are fed into the network to be compared, it becomes computationally too expensive for semantic textual similarity tasks. Some examples include finding the most similar pair of sentences in a dataset and clustering similar sentences.

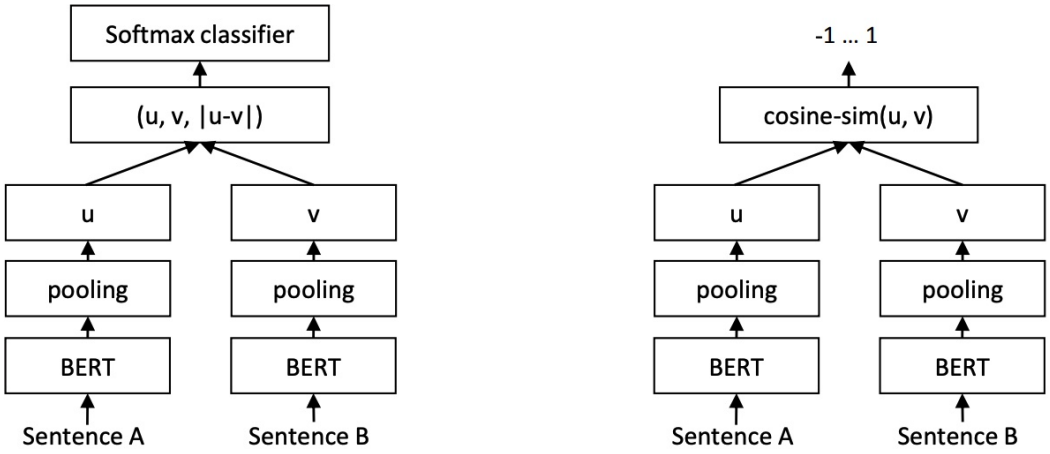

FIGURE 2.2: Figure taken from Reimers and Gurevych, 2019. Left: SBERT architecture with classification objective function. Right: SBERT architecture with regression objective function and at inference.

Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) is a modification of the pre-trained BERT network using a siamese or triplet network structure to create meaningful sentence embeddings. It aggregates the words in a sentence that are independently encoded by BERT using pooling to derive their

sentence embeddings. These embeddings can then be compared using a cosine similarity function. It can be optimized for regression or classification tasks.

2.3 Cosine Similarity

In natural language processing, cosine similarity is a metric used to measure the similarity between two documents. It is robust to scaling unlike some other distance measures such as Euclidean distance, where the distance would increase in proportion to the difference of the length of the two documents. In contrast, it measures the distance between two documents as the cosine of the angle of the two document vectors in a multidimensional space (Piuri et al., 2020). Furthermore, it is computationally inexpensive (Wang et al., 2019). Its formula is given by:

$$similarity(x, y) = cosine(\theta) = \frac{x \cdot y}{||x||||y||}$$

If the two vectors have opposite meaning, the cosine value will be close to -1. Moreover, if the cosine value is 0, the two vectors are orthogonal and have no match (Han, Kamber, and Pei, 2012). As the cosine value gets closer to 1, the similarity between the two vectors increase.

2.4 Cross-Entropy Loss

Cross-entropy loss is a cost function used in classification problems in machine learning to optimize the model during training.

In binary classification, where the number of classes, M, equals 2, binary cross-entropy is defined as:

$$-(y \log(p) + (1 - y) \log(1 - p))$$

In multiclass classification, i.e., $M > 2$, the loss is calculated separately for each class per observation and summed:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where y is the binary indicator (0 or 1) if prediction c is the correct classification for observation o, and p is the predicted probability observation o is of class c (Goodfellow, Bengio, and Courville, 2016).

Chapter 3

Related Work

Contextual neural embeddings are widely used to represent textual information across many domains. As explained in Chapter 2, they can be finetuned for a variety of domain-specific tasks. Since our task is essentially to create domain-specific sentence embeddings, we will focus on reviewing literature that use embedding-based methods for the candidate-job match task.

Two of the most similar studies (Lavi, Medentsiy, and Graus, 2021), (Nigam et al., 2021) in the literature involve the use of word and sentence embeddings. They both notice a significant improvement in their model performances compared to pre-trained BERT models. However, these studies manually label very large datasets and we do not have the resources to do so.

Lavi, Medentsiy, and Graus (2021) finetune a multilingual BERT model using the Siamese SBERT framework for the resume to vacancy match task on a large resume-vacancy pair dataset. They label 270k pairs of resume-vacancy pairs using their consultants' decisions. They evaluate their model based on both classification and regression tasks. Their finetuned model outperforms their supervised and unsupervised baselines.

Nigam et al. (2021) fine-tune BERT for the skill to competency classification task on their skill and competency group pair dataset. They extract 3k unique skills from 700k job requisitions and classify these skills into 40 competency groups. Additionally, within a competency group, they classify the skills as *core* or *fringe*. Their aim is to use these competency groups in place of the skills to increase the performance of similarity computations between employee-job pairs. Their finetuned model outperforms their baselines by capturing information about skills' ontology.

The candidate-job match task has not only been studied using text-based matching models, but also with relation-based matching models. Bian et al. (2020) uses an ensemble of the two methods to combine their merits for the task. Similarly, the future work of this project will aim to use combine knowledge graphs and word embeddings.

Chapter 4

Datasets

In this chapter, we describe the datasets used in this project and the challenges that they present.

4.1 Swisscom Datasets

4.1.1 Employee Skill Profiles

The main dataset used in this project is a Swisscom employee skills dataset that comprises of over 200k skill entries. Each row in the dataset contains a single skill text along with the employee number that has entered it into their profile and a self-rating (0-3) of the skill. The dataset is sourced from two platforms. Therefore, depending on the source, a self-rating of 0 can either mean the skill is not learned yet or the employee has not rated it. Before any preprocessing, there are over 10k profiles consisting of a median of 8 skills per profile.

Since these skills are entered manually by the company employees onto the company tool, the dataset presents several challenges. First, the dataset is multilingual, comprising of four languages, i.e., English, German, French, and Italian. Moreover, it contains skills at different levels of granularity, e.g., *data science* at a coarser granularity and *pandas* at a finer granularity, while referring to similar skills. The same skills are often represented differently across profiles using synonyms, e.g., *collaboration* and *teamwork*. In addition, the employee skill profiles are often incomplete due to the employees either not having entered any skills or having entered only a subset of their skills since it is a challenging task to give an exhaustive list of skills that one has acquired throughout years of education and experience. This is reflected in the number of skills written in the profiles and consequently in a small dataset

to learn from. In contrast, some employees may enter skills that are neither relevant for their current role nor for their future career goals.

4.1.2 Job Information

The supplementary dataset used in this project is a Swisscom job information dataset that comprises job titles along with their level, cluster, and description. Job levels indicate the seniority level of a job title. There are 7 job clusters in total, which are 'Infrastructure', 'Products, Marketing, Sales & Customer Care', 'Business Management', 'Business Analysis & Engineering', 'Project Management', 'Development', and 'Operations'. Job descriptions indicate the required skills that the employee would need to acquire for that role. Each job title is assigned a job cluster. However, the number of job titles contained under each cluster is imbalanced. The top two clusters, i.e., 'Development' and 'Products, Marketing, Sales & Customer Care', contain more than half of the job titles, leading to significant imbalance in the number of employee profiles in the employee skill profiles dataset.

4.2 External Dataset

The external dataset used in this project is an open-source dataset provided on Kaggle (PromptCloud, Datastock, 2017). This dataset serves as a common dataset to compare the performances of the BERT-based method to the knowledge graph-based method used in the Roche team. It contains a list of required skills for 22k technology jobs posted on Dice[1] job board. Only a subset of these postings has been used in this dataset after preprocessing and labeling, as explained in Chapter 5.

[1] https://www.dice.com/

Chapter 5

Methods

5.1 Data Preprocessing

In this section, we describe the methods we use to acquire the data used in the train, validation, and test sets.

5.1.1 Swisscom Datasets

Several data preprocessing steps are used to overcome the irregularities in the Swisscom employee skills dataset mentioned in Chapter 4. The data preprocessing steps are included in the final model only if they increase the model performance on the validation set. First, all unique skills are extracted and translated from German, French, and Italian to English using the company translation tool, and are replaced within the dataset. Then, the skills that were rated 0 are removed as they may indicate that the skill is not necessary for the employees' current role. The employees who only have a single skill in their profile are removed because a single skill cannot reflect the skill set of an employee and therefore may introduce noise to the data. Moreover, the skills that occur only once across employee profiles are removed since they may indicate very rare skills that may be irrelevant for the company.

After all the preprocessing steps, the dataset consists of 9.3k employee profiles, which contain a median of 11 skills. Over 50% of the skills are entered by the employees in the 'Development' and 'Products, Marketing, Sales & Customer Care' clusters, which is proportional to the number of employees working there. The dataset is split using an 80% - 10% - 10% ratio into train, validation, and test sets, respectively. Stratified sampling is used to ensure equal ratios of profiles from each job cluster in each set.

5.1.2 External Dataset

Due to the various origins of the job postings on the Dice.com job board, the job postings in the original dataset do not contain cluster labels. Therefore, after sorting the job titles based on frequency of occurrence, the job postings under the top 6 job titles have been selected, and the job titles have been assigned as their cluster labels. The cluster labels are 'software', 'java', '.net', 'business', 'project', 'network'.

After all the preprocessing steps, the dataset consists of 7,148 job roles. To keep consistency between the projects in the Swisscom and Roche teams, the same final external dataset[1] is used, which is split using an 75% - 12.5% - 12.5% ratio into train, validation, and test sets, respectively. Compared to the Swisscom employee skill profiles dataset, this dataset is more balanced in terms of the job cluster labels. There are 1,642 samples in the *software*, 1,432 samples in the *java*, 1,240 samples in the *.net*, 1,060 samples in the *business*, 996 samples in the *project*, and 778 samples in the *network* cluster.

5.2 Employee Profile Representation

As our objective is to create meaningful representation of textual information in employee skill profiles to compute matching scores between employees and potential future roles, we begin by constructing the employee profile representation. We use BoW and TF-IDF with multinomial logistic regression as baselines. Since BERT can understand context and ambiguity in language due to its attention mechanism, we use it to solve the granularity, synonyms, and incompleteness problems in the Employee Skill Profiles dataset as mentioned in Chapter 4. Aligned with previous work (Lee et al., 2019), (Alsentzer et al., 2019), we fine-tune the pre-trained BERT model on our specialty corpora to improve the performance of our domain-specific model.

Unlike previous work in employee-vacancy matching (Lavi, Medentsiy, and Graus, 2021), we do not have the resources to label the employee-role pairings as having a "good" or "bad" matching. However, we do have job cluster labels for each job title, which we use the pre-trained BERT model to fine-tune on. This not only enables the BERT model to learn the collection of skills that are relevant to each job cluster, but it also acts as a heuristic method

[1] The preprocessed dataset may be found at: https://tinyurl.com/4twcnjem

to evaluate the quality of the embeddings.

We apply the same methods to both the internal and external datasets.

5.2.1 Baselines

For the baselines, we utilize BoW as the employee skills are given as a list of skills, and TF-IDF as it can emphasize the skills that are unique to job roles and diminish the effect of the most common skills across all profiles. To classify each profile into job clusters, we use a stochastic gradient descent classifier with cross-entropy loss. The fast training and evaluation time allow us to tune the alpha hyperparameter, which is the constant that multiplies the regularization term, where a larger value leads to stronger regularization. Both the BoW and TF-IDF models are tuned with 15 evenly spaced alpha hyperparameters, where the range of values are $[10^{-6}, 10^{-1}]$, $[10^{-8}, 10^{-3}]$, respectively.

5.2.2 BERT

BERT can achieve state-of-the-art models for a wide variety of natural language processing tasks by simply fine-tuning using an additional output layer (Devlin et al., 2018). The pre-trained model[2] uses masked language modeling and next-sentence prediction objectives on BookCorpus[3], a dataset of 11k unpublished books, and English Wikipedia.

To fine-tune the pre-trained BERT model for our downstream task, we use a sequence classification head on top of the pre-trained BERT model. More specifically, we append a linear layer on top of the pooled model output. We use cross-entropy loss to optimize on the job cluster classification task. We fine-tune our model for 5 epochs, using a batch size of 8, which is the largest size our system allows even though the lowest size the authors recommend for fine-tuning is 16. As the BERT model has a maximum limit of 512 tokens, we trim the employee skills in each profile to 512 tokens.

[2] We used the `bert-base-uncased` model from the HuggingFace library (Wolf et al., 2019).
[3] https://yknzhu.wixsite.com/mbweb

5.3 Calculating Similarity

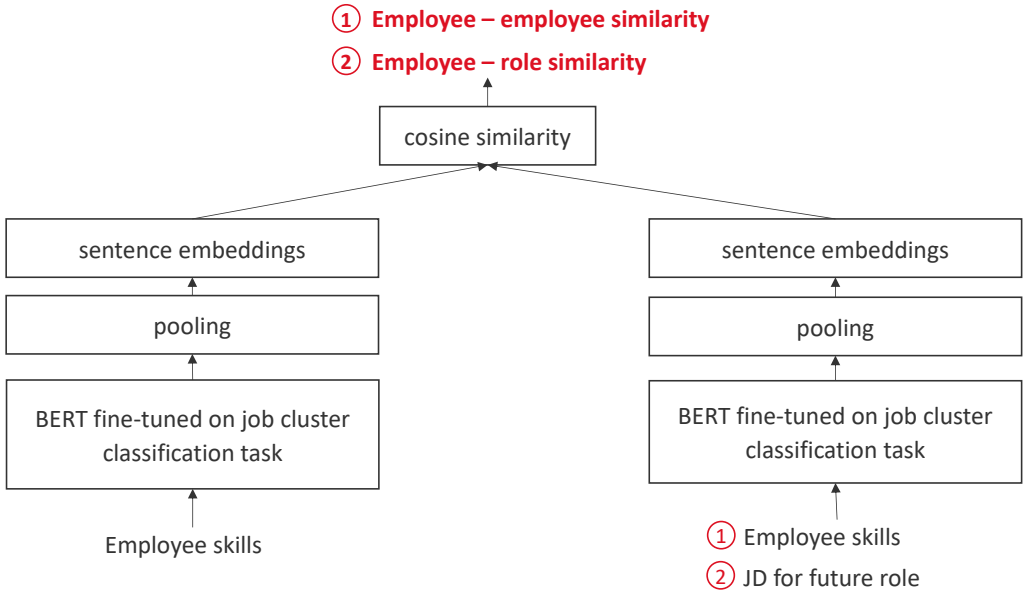

FIGURE 5.1: Model architecture used in the project.

As explained in Chapter 2, the SBERT model (Lavi, Medentsiy, and Graus, 2021) aggregates the words in a sentence that are independently encoded by BERT using mean pooling. Thus, it allows for efficient computation of cosine similarity scores between sentences, i.e., two employee profiles as well as employee profiles and potential future job roles. Consequently, it can output the most similar employee profiles and the most similar employee - job role matches.

We used SBERT[4] on the best performing fine-tuned BERT model on the job cluster classification task to obtain the sentence embeddings of each employee skill profile. The SBERT embeddings also allow for efficient clustering of the employee profiles, which we then use to visualize using the dimensionality reduction technique t-distributed stochastic neighbor embedding (t-SNE). This enables us to evaluate the SBERT representation of the employee skill profiles. Furthermore, we visualize the employee skill profiles using interactive plots[5] and inspect the profiles that are closest to each other in the 2-dimensional space. Finally, we select a subset of examples from the

[4]We used the SentenceTransformers library (Lavi, Medentsiy, and Graus, 2021).
[5]We used the Bokeh library.

Swisscom job information dataset and compare them with a subset of employee skill profiles with varying degrees of similarity.

For the external experiments, due to the lack of an open-source employee skills dataset, we apply the abovementioned procedure to compute similarity between pairs of skills written under job roles only.

Chapter 6

Results

In this chapter, we present the results of the experiments conducted on the Swisscom datasets as well as the external dataset.

6.1 Swisscom Results

6.1.1 Job Cluster Classification

Most preprocessing steps as explained in Chapter 5 increase the performance on the job cluster classification task. The best performance is achieved by translating the skills to English, removing the skills with a skill rating of 0, and removing profiles with a single skill. In particular, removing skills with a skill rating of 0 and removing profiles with a single skill increase the model performance by 3.1%. On the other hand, removing skills with a single occurrence among all employee profiles reduces the performance. This is likely due to the abundance of synonyms used across profiles to represent similar skills, which results in a large number of singly occurring skills.

Fine-tuning BERT on the employee skill profiles dataset on the job cluster classification task where there are 7 clusters, we obtain the best loss on the validation set using 3 epochs. The results of the best models are shown in Tables 6.1 and 6.2.

As expected, BoW and TF-IDF are unable to represent employee skill profiles as depicted in the t-SNE plots in Figure 6.1. On the other hand, SBERT embeddings produce clear clusters of employee skill profiles with respect to the job clusters the employees belong to.

Looking into the SBERT t-SNE plot in detail (Figure 6.1 - top right), the proximity of the job clusters help us evaluate the quality of learning. For

Model	Weighted-F1	Weighted-Precision	Weighted-Recall
BoW+LR	$b_1 (\approx 0.6)$	$b_1 + 0.0108$	$b_1 + 0.0218$
TF-IDF+LR	$b_1 + 0.0308$	$b_1 + 0.0360$	$b_1 + 0.0474$
BERT	$b_1 + 0.0482$*	$b_1 + 0.0471$*	$b_1 + 0.0548$*

TABLE 6.1: Job cluster classification test set performances (weighted-f1, -precision, and -recall) of different embedding models on the Swisscom employee skill profiles dataset. Relative results are shown due to confidentiality.

Model	Macro-F1	Macro-Precision	Macro-Recall
BoW+LR	$b_2 (\approx 0.5)$	$b_2 + 0.0828$	$b_2 - 0.0294$
TF-IDF+LR	$b_2 + 0.0418$	$b_2 + 0.1157$*	$b_2 + 0.0117$
BERT	$b_2 + 0.0748$*	$b_2 + 0.1022$	$b_2 + 0.0579$*

TABLE 6.2: Job cluster classification test set performances (macro-f1, -precision, and -recall) of different embedding models on the Swisscom employee skill profiles dataset. Relative results are shown due to confidentiality.

example, the *Business Analysis & Engineering* cluster appears in between the *Development* and *Business Management* clusters, which validates our model as the employees in the *Business Analysis & Engineering* cluster need to acquire both technical and business skills. On the other hand, the *Products, Marketing, Sales & Customer Care* cluster appears farthest from the *Development* cluster. This provides another validation as the employees in these two clusters need to acquire highly dissimilar skills. On the other hand, the employee profiles in the smallest cluster *Project Management* do not form a clear cluster. However, it can be argued that given more data, the BERT model could cluster *Project Management* employee profiles as well, as the two majority classes form much clearer clusters compared to the other job clusters.

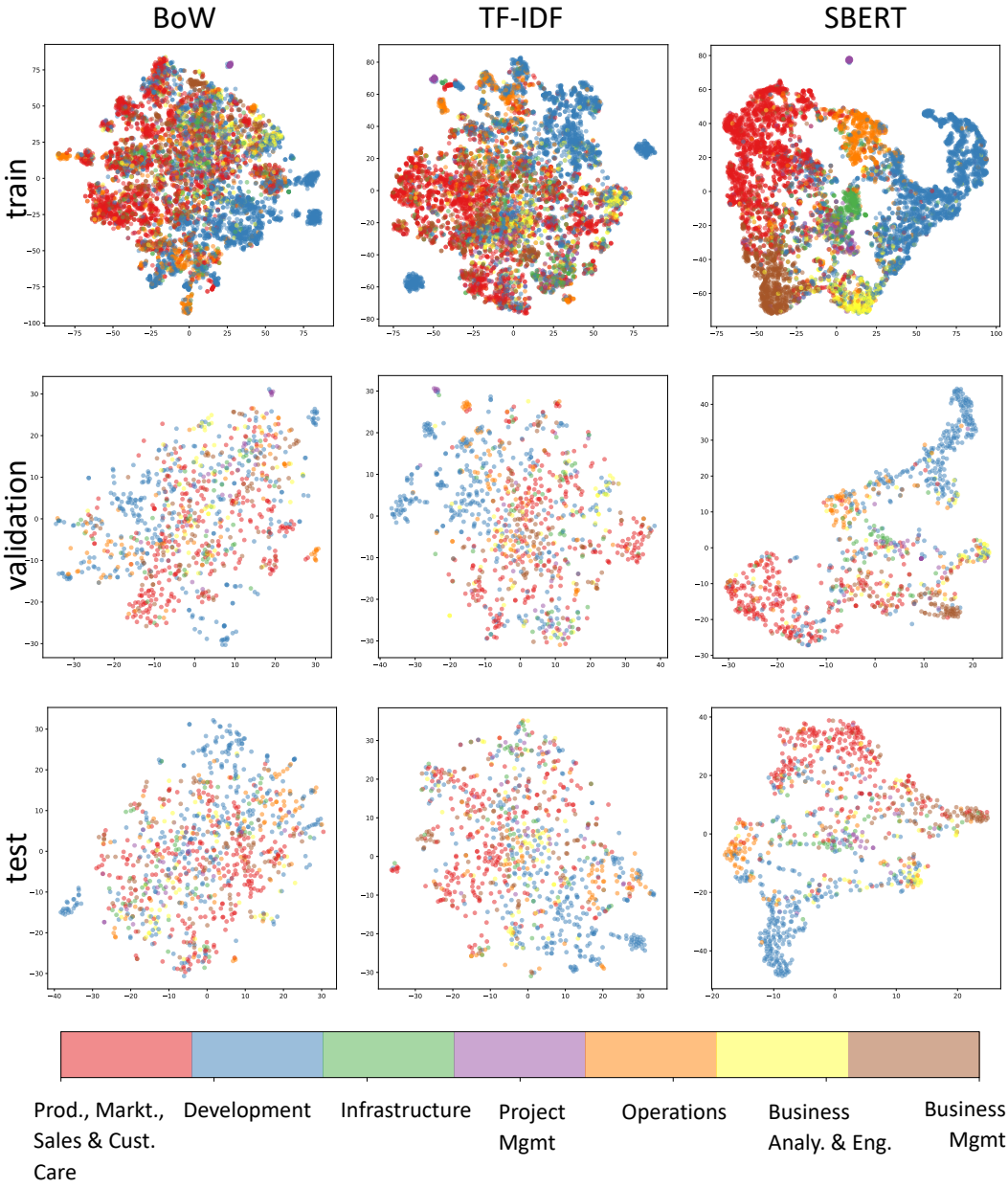

FIGURE 6.1: t-SNE plot of Swisscom employee profiles. Left column: BoW; middle column: TF-IDF; right column: SBERT. Top row: train set; middle row: validation set; bottom row: test set. Legend shows job cluster label colors.

6.1.2 Applications

Using the computed SBERT embeddings of the employee skill profiles and the job roles, we compute two types of similarities. The first is the employee - employee similarity to validate our model by manually checking pairs of employee profiles. The second is employee - job role similarity to evaluate the value of our model.

Due to confidentiality, for the Swisscom examples shown in this report, we create synthetic employee profiles using skills from employee profiles with the same job title in the dataset. Moreover, instead of showing the required skills in the internal job information dataset, we show the skills from the open-access job descriptions[1].

Table 6.3 shows an example comparing a DevOps Engineer profile to the DevOps Engineer job role. This comparison serves as a sanity check for our model. The cosine similarity score is 0.9370, which validates that our model is able to predict high similarity between similar pairs of employee profiles and job roles.

	Title	Skills
Employee	DevOps Engineer	DevOps; Git; LINUX; Java; HTML; Shell; JavaScript; Database; Agile; Development; CSS; English; French; Software Engineering; Docker; Python
Job	DevOps Engineer	Developing, engineering and operating skills for monitoring applications in a networking environment; Strong knowledge and experience with Fault Management and RCA tools (such as VMware SMARTS) and Performance Reporting tools (such as VMware Watch4Net); Good knowledge of the cloud environment/technologies (Kubernetes, Docker); Knowledge of a programming language such as Python, Java or JavaScript

TABLE 6.3: Example of a DevOps Engineer profile - DevOps Engineer job role comparison. The cosine similarity score is 0.9370.

[1] https://www.swisscom.ch/en/about/career/vacancies.html

Table 6.4 shows an example comparing completely different pairs of list of skills, i.e., the same synthetic DevOps Engineer profile compared with the Senior Digital Media Planner job role. This time, the cosine similarity score is -0.0693, which further validates our model, and shows that our model is able to predict low similarity between dissimilar pairs of employee profiles and job roles.

	Title	Skills
Employee	DevOps Engineer	DevOps; Git; LINUX; Java; HTML; Shell; JavaScript; Database; Agile; Development; CSS; English; French; Software Engineering; Docker; Python
Job	Senior Digital Media Planner	5+ years of online work experience with an agency or client; Technical understanding of the requirements of the online world and proficiency with at least one AdManager; In-depth know-how and passion for data-driven digital marketing; Experience in independently setting up and managing projects and experience in an online agency an asset

TABLE 6.4: Example of a DevOps Engineer profile - Senior Digital Media Planner job role comparison. The cosine similarity score is -0.0693.

	Title	Skills
Employee	DevOps Engineer	DevOps; Git; LINUX; Java; HTML; Shell; JavaScript; Database; Agile; Development; CSS; English; French; Software Engineering; Docker; Python
Job	DevOps Agile Coach	Several years of experience in a leadership function in a technical environment; Ability to lead and drive large scale change within an international engineering and services organization; Experienced in agile development methods and DevSecOps Culture; SAFe; DevOps; Personnel management

TABLE 6.5: Example of a DevOps Engineer profile - DevOps Agile Coach job role comparison. The cosine similarity score is 0.7120.

Table 6.5 shows an example that depicts the value of our model. A DevOps Engineer could potentially apply for a DevOps Agile Coach role in the

future. However, despite fulfilling the technical skills, the employee lacks the leadership and management skills required for the role. Therefore, the cosine similarity score is 0.7120.

Even though the model can accurately predict the similarity between highly technical jobs, it struggles with more soft-skill based jobs. Table 6.6 shows an example comparing a Leader Consulting profile to a Customer Service profile. The two profiles are contained within *Development* and *Products, Marketing, Sales & Customer Care* clusters, respectively. Although these two profiles are contained in entirely different clusters and do not acquire the same skills, their computed similarity score is quite high. This is partly due to an irrelevant skill, i.e., *personal training*, contained in both profiles that are uncommon in the rest of the dataset. In addition, both profiles acquire soft skills and language skills that are not necessarily specific to their job titles.

	Title	Skills
Employee	Leader Consulting	Empathy; Coaching; Agile; Personal Training; Application Development Lifecycle; Agile Coaching; Leadership; Business Development; Sales; Lifelong Learning; English; French; Strive for Excellence
Employee	Customer Service	IT system knowledge; Language Learning; Lifelong Learning; Fitness; conscientious; Customer Support; Call centers; Adaptable; number-oriented; cultivated; willing to learn; German; team skills

TABLE 6.6: Example of a Leader Consulting profile - Customer Service profile comparison. The cosine similarity score is 0.8922.

6.2 Academic Results

6.2.1 Job Cluster Classification

Fine-tuning BERT on the employee skill profiles dataset on the job cluster classification task where there are 6 clusters, we obtain the best loss on the validation set using 3 epochs. The results are shown on Tables 6.7 and 6.8. Since the external dataset consists of mostly very technical keyword-type skills, the BoW and TF-IDF models perform better on this dataset than on the Swisscom employee skills dataset. However, Figure 6.2 still shows much

clearer clusters with the SBERT embeddings than the baselines. Thus, the fine-tuned BERT model is able to learn which type of employee skill profiles belong to which job cluster.

Model	Weighted-F1	Weighted-Precision	Weighted-Recall
BoW+LR	0.7196	0.7222	0.7226
TF-IDF+LR	0.6971	0.7003	0.6980
BERT	0.7250*	0.7294*	0.7271*

TABLE 6.7: Job cluster classification test set performances (weighted-f1, -precision, and -recall) of different embedding models on the Dice skills for job roles dataset.

Model	Macro-F1	Macro-Precision	Macro-Recall
BoW+LR	0.7287	0.7295	0.7340
TF-IDF+LR	0.7038	0.7091	0.7030
BERT	0.7373*	0.7350*	0.7474*

TABLE 6.8: Job cluster classification test set performances (macro-f1, -precision, and -recall) of different embedding models on the Dice skills for job roles dataset.

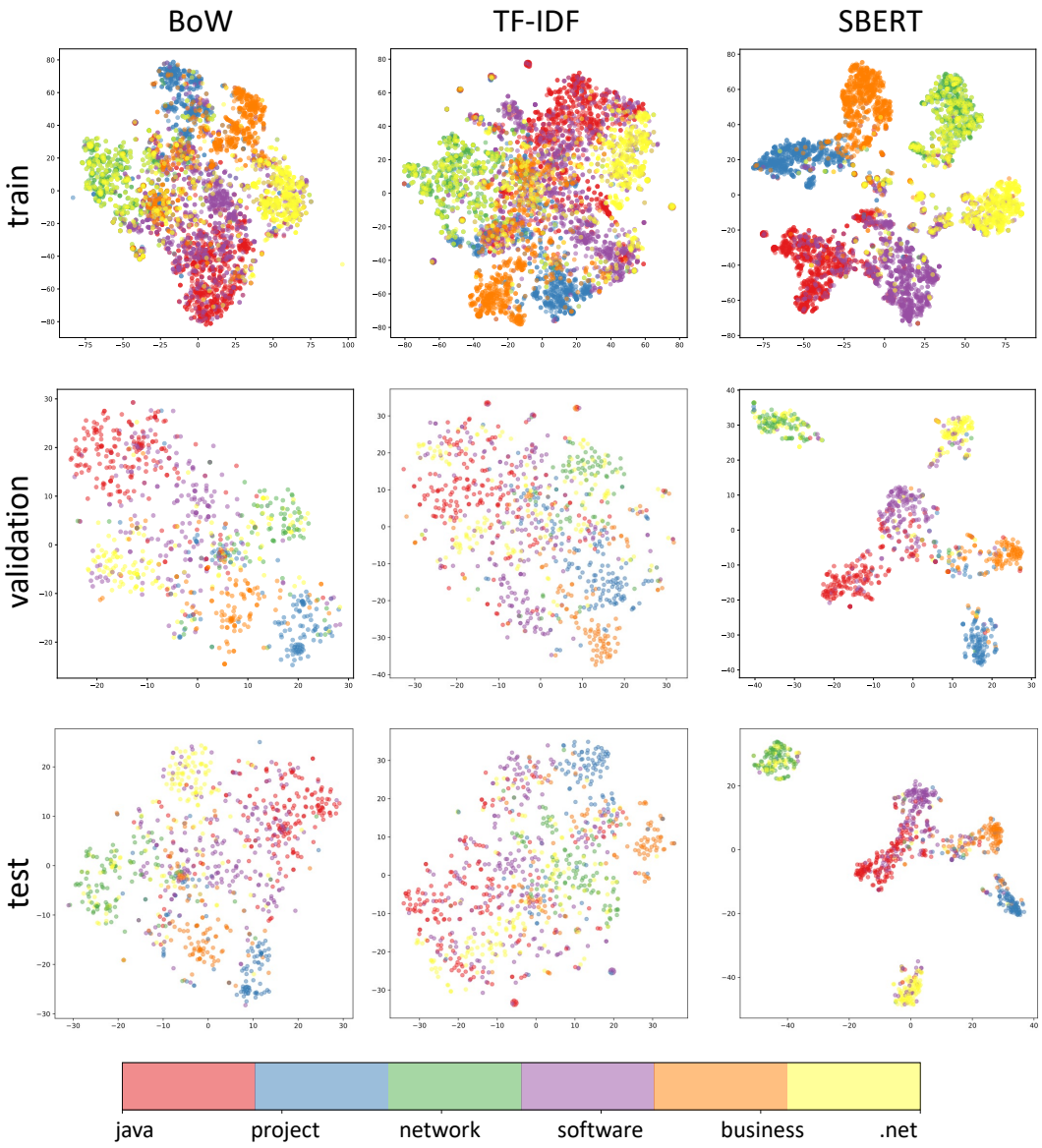

FIGURE 6.2: t-SNE plot of Dice job roles. Left column: BoW; middle column: TF-IDF; right column: SBERT. Top row: train set; middle row: validation set; bottom row: test set. Legend shows job cluster label colors.

6.2.2 Applications

Using the computed SBERT embeddings of the required skills in job roles, we compute cosine similarity between job roles on the external dataset. We check a subset of pairs of job roles with their similarity scores to evaluate the quality of our model.

Table 6.9 shows an example comparing a Software Engineer/Linux Network Programmer role to a Tech Support/Network Admin role. Our model is able to detect the similarity of skills, specifically network skills, required for both job roles and outputs a high similarity score of 0.9144.

	Title	Skills
Job	Software Eng. / Linux Network Programmer	Java C++ Webservices rest API restful tcp tcp/ip ip ipv5 ipv6 linux network stack sockets multi-threading testing docker vmware containers
Job	Tech Support / Network Admin	Networking, TCP/IP, Routing & Switching (OSPF / BGP / VLAN / STP), Cisco, Checkpoint, Juniper (Netscreen), Fortinet product, Authentication Protocols a plus (Radius / TACACS)

TABLE 6.9: Example of a Software Engineer/Linux Network Programmer - Tech Support / Network Admin role comparison. The cosine similarity score is 0.9144.

Table 6.10 shows an example comparing a Lead Business Analyst role to a Java Developer role. Our model is able to detect the dissimilarity between the required skills for the two roles and outputs a high dissimilarity score of -0.4123.

	Title	Skills
Job	Lead Business Analyst	Business Analyst, Project Manager, Banking, Documentation, Processes, Procedures, Liaison
Job	Java Developer	Java J2ee Spring Hibernate, Struts, EJB, Web Service, SVN, SDLC, HTML UI, Javascript, Node.JS, Angular

TABLE 6.10: Example of a Lead Business Analyst - Java Developer role comparison. The cosine similarity score is -0.4123.

Chapter 7

Discussion

Looking at the experiment results, our SBERT model can predict the similarity between employee-employee and employee-job pairs fairly well, especially in technical roles. However, when it comes to soft-skill based roles, the quality of the learning drops and it gets difficult for the model to distinguish between roles in different job clusters. To counter this shortcoming of our model, we need additional data. For example, a dataset with hierarchical information of soft skills, hard skills, and language skills could benefit the model in making accurate distinction between such profiles. Moreover, the information of agile-based roles of employees could help the model learn the soft skills required for such roles. In addition, the employee skill profiles can be enriched using skills from the job descriptions to further overcome its incompleteness challenge.

Potential improvements for the methods include further data preprocessing to remove irrelevant skills for Swisscom careers. Moreover, fine-tuning SBERT using different loss functions, such as multiple negatives ranking loss or contrastive loss, can give the model another opportunity to learn what type of profiles and roles are similar to each other, in addition to fine-tuning on the job cluster classification task. Ultimately, an ensemble of knowledge graphs and word embeddings can leverage the representation advantages of both methods.

Chapter 8

Conclusion

In this work, we introduce a Sentence-BERT model that can compute candidate-job role similarity using cosine similarity, and consequently find the best-fit candidate for a role. The current literature only presents methods that include manual labeling of hundreds of thousands of employee-role pairs. Despite not having the labels for our downstream task, which is the similarity between employee profiles and job roles, our proposed model can compute meaningful similarity scores thanks to fine-tuning the pre-trained BERT model on the job cluster classification task. The model successfully shows that there is predictive power in the employee skill profiles and that this area is worth further exploring.

The contextual BERT embeddings alleviate the incompleteness of profiles using the context of the existing skills as well as the difference in wordings used for the same skills across profiles and at different levels of granularity. Additionally, we present several data preprocessing methods to further overcome the irregularities in the dataset. The t-SNE plots of employee skill profiles, and the employee-employee and employee-job role comparison examples validate that our model can construct meaningful embeddings and compute accurate similarities. The evaluation on external data further shows that the model works especially well for technical roles. However, the model can benefit from additional data that captures the hierarchical structure of skills for soft skill-based roles.

Bibliography

Alsentzer, Emily et al. (June 2019). "Publicly Available Clinical BERT Embeddings". In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 72–78. DOI: 10.18653/v1/W19-1909. URL: https://aclanthology.org/W19-1909.

Bian, Shuqing et al. (2020). "Learning to Match Jobs with Resumes from Sparse Interaction Data using Multi-View Co-Teaching Network". In: *CoRR* abs/2009.13299. arXiv: 2009.13299. URL: https://arxiv.org/abs/2009.13299.

Devlin, Jacob et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Han, Jiawei, Micheline Kamber, and Jian Pei (2012). *Data mining concepts and techniques, third edition*. URL: http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm_hrd_title_0?ie=UTF8&qid=1366039033&sr=1-1.

Lavi, Dor, Volodymyr Medentsiy, and David Graus (2021). "conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers". In: *CoRR* abs/2109.06501. arXiv: 2109.06501. URL: https://arxiv.org/abs/2109.06501.

Lee, Jinhyuk et al. (2019). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *CoRR* abs/1901.08746. arXiv: 1901.08746. URL: http://arxiv.org/abs/1901.08746.

Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].

Nigam, Amber et al. (2021). *Skill{BERT}: "Skilling" the {BERT} to classify skills!* URL: https://openreview.net/forum?id=TaUJl6Kt3rW.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162.

Piuri, V. et al., eds. (2020). Hybrid Computational Intelligence for Pattern Analysis and Understanding. 978-0-12-823268-2. Elsevier, p. 306.

PromptCloud, Datastock (2017). *U.S. Technology Jobs on Dice.com*. data retrieved from Kaggle, https://www.kaggle.com/PromptCloudHQ/us-technology-jobs-on-dicecom.

Reimers, Nils and Iryna Gurevych (Nov. 2019). "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. URL: https://arxiv.org/abs/1908.10084.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

Wang, Bin et al. (2019). "Evaluating word embedding models: methods and experimental results". In: *APSIPA Transactions on Signal and Information Processing* 8.1. ISSN: 2048-7703. DOI: 10.1017/atsip.2019.12. URL: http://dx.doi.org/10.1017/ATSIP.2019.12.

Wolf, Thomas et al. (2019). "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *CoRR* abs/1910.03771. arXiv: 1910.03771. URL: http://arxiv.org/abs/1910.03771.

Zahidi, Saadia et al. (2020). *The Future of Jobs Report 2020*. Tech. rep. World Economic Forum Platform for Shaping the Future of the New Economy and Society.