Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.070813

ARTICLE

Classification of Job Offers into Job Positions Using O*NET and BERT Language Models

Lino Gonzalez-Garcia*, Miguel-Angel Sicilia and Elena García-Barriocanal

Computer Science Department, Universidad de Alcalá, Madrid, 28801, Spain

*Corresponding Author: Lino Gonzalez-Garcia. Email: lino.gonzalez@uah.es

Received: 24 July 2025; Accepted: 23 October 2025; Published: 09 December 2025

ABSTRACT: Classifying job offers into occupational categories is a fundamental task in human resource information systems, as it improves and streamlines indexing, search, and matching between openings and job seekers. Comprehensive occupational databases such as O*NET or ESCO provide detailed taxonomies of interrelated positions that can be leveraged to align the textual content of postings with occupational categories, thereby facilitating standardization, cross-system interoperability, and access to metadata for each occupation (e.g., tasks, knowledge, skills, and abilities). In this work, we explore the effectiveness of fine-tuning existing language models (LMs) to classify job offers with occupational descriptors from O*NET. This enables a more precise assessment of candidate suitability by identifying the specific knowledge and skills required for each position, and helps automate recruitment processes by mitigating human bias and subjectivity in candidate selection. We evaluate three representative BERT-like models: BERT, RoBERTa, and DeBERTa. BERT serves as the baseline encoder-only architecture; RoBERTa incorporates advances in pretraining objectives and data scale; and DeBERTa introduces architectural improvements through disentangled attention mechanisms. The best performance was achieved with the DeBERTa model, although the other models also produced strong results, and no statistically significant differences were observed across models. We also find that these models typically reach optimal performance after only a few training epochs, and that training with smaller, balanced datasets is effective. Consequently, comparable results can be obtained with models that require fewer computational resources and less training time, facilitating deployment and practical use.

KEYWORDS: Occupational databases; job offer classification; language models; O*NET; BERT; RoBERTa; DeBERTa

1 Introduction

Matching job offers to occupational categories is essential for effectively filtering opportunities for job seekers. This process ensures accurate classification of vacancies according to standardized occupational frameworks such as O*NET[1] or ESCO[2]. By aligning job descriptions with recognized taxonomies, recruitment systems can more precisely identify relevant skills, qualifications, and responsibilities, thereby improving the precision of search and recommendation mechanisms. This facilitates the discovery of suitable positions for candidates and improves the ability of employers to reach qualified applicants. In addition, it contributes to labor market analysis by providing structured data to monitor employment trends and skill demands.

[1] https://www.onetonline.org/ (accessed on 20 October 2025)
[2] https://esco.ec.europa.eu/ (accessed on 20 October 2025)

However, the heterogeneity and unstructured nature of job offer descriptive texts pose a significant challenge to this task. This has been addressed in previous work using diverse methods of automation of the matching. The use and fine-tuning of pre-trained language models (LMs) such as BERT reduces the computational cost of training and leverages the capabilities of general-purpose LMs while accounting for textual variation. Previous studies have already applied those language models for the classification of job offers, but they typically focus on a single occupational category or employ a single language model in the classification pipeline. For example, Skondras et al. [1] use only the BERT model for job-title classification; Maitra et al. [2] use DistilBERT and compare it against traditional machine learning models for job-title classification; Clavié et al. [3] perform a binary classification to determine whether a position is suitable for a recent graduate; and Beristain et al. [4] develop a job classifier restricted to the information technology domain. In contrast, we conducted a systematic, comparative evaluation of the main models from the BERT family (BERT, RoBERTa, and DeBERTa), employing them as multiclass classifiers to assign job offers to their primary occupational category. As a reference, we use one of the largest and most comprehensive occupational databases, O*NET, which covers the entire labor market across 23 occupational categories and 1,016 occupations. For this task, we fine-tune the models on a dataset of job postings collected via the official job-search tool offered by O*NET. The dataset includes the job title, job description, and the occupational family label assigned according to the O*NET classification.

Results show that the most suitable model for this classification task is DeBERTa, achieving a macro F1-score of 0.767 and demonstrating high accuracy across most occupational categories. An analysis by class indicates that some categories are classified with particularly high precision, such as Food Preparation and Serving Related, which achieved an F1-score of 0.902, while others exhibit lower precision, such as Management, with an F1-score of 0.645. This pattern likely reflects similarities among neighboring categories in the O*NET taxonomy, which complicate classification when job postings lack detailed or precise descriptions.

The rest of this paper is structured as follows. Section 2 presents an overview of occupational databases and the application of language models to job offer classification tasks. Section 3 details the construction of the training dataset and the fine-tuning process applied to the various language models. Section 4 reports the experimental results obtained, and finally, Section 5 outlines the conclusions and directions for future work.

2 Background

2.1 Occupational Databases

Occupational databases are essential tools for collecting, organizing, and disseminating information related to diverse occupations. These databases systematically organize data that describe job roles, responsibilities, required skills, employment conditions, and occupational health risks. Their purpose is multifaceted: they serve job seekers, employers, researchers, and policymakers by providing essential information on labor market dynamics and occupational health [5–7]. These resources enable informed decision-making for individuals seeking employment or career advancement, while also assisting employers with workforce planning and regulatory compliance.

One of the most notable global occupational databases is the Occupational Information Network (O*NET), managed by the U.S. Department of Labor. O*NET offers a comprehensive repository of information on more than 1000 occupations, categorized into 23 occupational families, detailing job requirements, worker attributes, and the characteristics of various job roles [8,9]. Each occupation is described using a standardized set of descriptors, which include knowledge, skills, interests, abilities, and work activities, allowing systematic analysis of job characteristics across different sectors [9]. In addition, O*NET provides

other employment-related tools, such as its job-search portal (https://www.careeronestop.org/, accessed on 20 October 2025), which offers a wide range of job listings categorized by occupation and location.

Complementary to O*NET, other occupational databases, such as the European Skills, Competences, Qualifications, and Occupations (ESCO) database, offer a complementary perspective. ESCO aims to establish a common language for job titles, skills, and qualifications across Europe, facilitating labor market analysis and workforce development. Similar to O*NET, ESCO categorizes occupations and their associated skills, but does so within the context of the European labor market, making it an essential resource for understanding job dynamics in that region.

Finally, it is important to highlight the integration of multiple occupational databases as a strategy to improve content and utility. For example, Babiuk et al. [10] demonstrated how combining data from the Canadian Career Handbook and O*NET can facilitate more complex occupational health queries, while Sreerambatla [11] explored the synthesis of O*NET and ESCO data to create a more comprehensive resource for job matching and career planning.

The use cases for these occupational databases are varied and highly beneficial for improving employability and optimizing job search strategies. For example, databases like O*NET allow potential candidates to match their personal skills and qualifications with job requirements, effectively bridging the gap between their capabilities and employer expectations [7,12]. On the other hand, employers utilize these databases to refine job descriptions, select suitable candidates, and develop training programs, thereby enhancing overall workplace efficiency and satisfaction [5].

2.2 Language Models in the Classification of Job Offers

The use of language models for job offer classification has gained significant momentum in recent years, driven by advances in natural language processing (NLP) and machine learning. These models facilitate the extraction, analysis, and categorization of job-related information from unstructured text, thereby enabling efficient matching between job postings and candidate profiles. Using state-of-the-art NLP techniques in conjunction with comprehensive occupational databases such as O*NET and ESCO, organizations can streamline recruitment processes and improve employment outcomes.

The integration of established occupational databases like O*NET and ESCO into classification pipelines plays a crucial role. When incorporated into NLP models, these databases can be used to fine-tune classification algorithms, thereby improving accuracy and ensuring that job postings align with candidate profiles and meet skill and qualification standards in the labor market.

Some approaches to matching job offers and positions use structured descriptions as input, e.g., using similarity matching of skill sets. However, we are interested here in matching that uses the text of the job offer without the need for preprocessing or elaboration.

Some previous work has used BERT models (Bidirectional Encoder Representations from Transformers) for the task. BERT, developed by Google, has emerged as a groundbreaking model due to its ability to understand context and semantic nuances within textual data, making it particularly suitable for analyzing job descriptions. The advantage of using BERT lies in its architecture, which processes text bidirectionally. Unlike traditional models that analyze text sequentially, BERT reads entire sentences or paragraphs at once, enabling it to capture relationships and contextual clues that are vital for accurate classification. This bidirectional understanding can provide insights into which skills, qualifications, and responsibilities are explicitly mentioned or implied in job postings, allowing for better alignment with potential applicants' profiles.

Skondras et al. [1] discuss plans to enhance job description classification by training BERT variants specifically on resume data, addressing potential misclassifications through another classification layer designed to discern specific characteristics in resumes. This highlights BERT's adaptability to focused datasets to improve the accuracy of job classification tasks.

Maitra et al. [2] use an enhanced BERT-based model for the multi-label classification of job descriptions into 137 different job titles. In this study, only the BERT model is used, whereas in our work, we compare this model with more advanced alternatives such as RoBERTa and DeBERTa.

Clavié et al. [3] use ULMFiT, DeBERTaV3, davinci-002, davinci-003 and GPT-3.5-turbo to classify job offers and determine whether they are appropriate for a graduate or entry-level position. A binary classification model is proposed that is much simpler than ours, which is multiclass, since it must classify the job offer among all the occupational families defined by O*NET.

Alonso et al. [12] use SentenceTransformers [13], a BERT-based framework, to vectorize job descriptions and subsequently compare them with O*NET entities in order to detect the Abilities, Knowledge, Skills, Work Activities, Tools Used, Task Ratings, and Technology Skills present in each job, thereby determining the most closely matching O*NET occupation.

Beristain et al. [4] develop a model to classify a job offer into a job position but only for information technology areas, since our model classifies the job offer for any O*NET occupational family.

In a more specialized endeavor, the introduction of JobBERT served to enhance job title understanding by integrating skill co-occurrence information from job descriptions, indicating a nuanced application of BERT-centric architectures in the realm of job title normalization [14]. This adaptation exemplifies how fine-tuning BERT can yield significant improvements in classification tasks specific to employment contexts.

Language models have been used in the context of occupational databases for other tasks different from the ones addressed here. For example, Xu et al. [15] use BERT models to predict the automation of O*NET and ESCO occupations, while Alonso et al. [16] leverage O*NET and ESCO as knowledge bases to refine and improve the responses of a ChatGPT-based chatbot specialized in the labor market.

3 Materials and Methods

3.1 Job Offer Database

To obtain the dataset used in this study, we developed a Python scraper [17] using the `BeautifulSoup` library. This scraper accessed the job search engine provided by the website (https://www.careeronestop.org/, accessed on 20 October 2025) to retrieve job listings for each of the 1016 occupation codes defined by O*NET. Subsequently, each result was categorized into the corresponding job family associated with the occupation code used in the query. A maximum of 3000 job postings were retrieved per occupation, resulting in a total of 692,600 job offers, all written in English. The distribution of these offers across employment families is shown in Table 1. We also observe that job offers were successfully extracted for all occupation families except for the Military Specific family. Consequently, this family was excluded from the analysis, resulting in a final dataset comprising 22 occupation families and 997 distinct occupations.

The dataset exhibits marked class imbalance across occupational families, with Healthcare Practitioners and Technical for the largest number of postings (72,353) and Legal the fewest (11,683). In addition, the postings-per-occupation density varies substantially across families (e.g., 2565 in Building and Grounds Cleaning and Maintenance vs. 355 in Production), suggesting that some occupations are represented by very few postings, which may hinder the detection of these minority occupations. Given this imbalance, we adopt macro F1-score as the primary evaluation metric. Unlike micro F1-score and weighted F1-score, both of which are dominated by majority classes, macro F1-score computes the F1-score independently for

each occupational family and then averages them with equal weight. This makes the metric sensitive to performance on minority families and better aligned with our objective of achieving balanced classification across the taxonomy.

Table 1: Distribution of job offers by O*NET occupational family

O*NET occupational family	O*NET occupations	Number of offers
(Class 0) Management	59	48,452
(Class 1) Business and Financial Operations	50	43,284
(Class 2) Computer and Mathematical	38	31,290
(Class 3) Architecture and Engineering	59	47,838
(Class 4) Life, Physical, and Social Science	66	43,102
(Class 5) Community and Social Service	18	16,572
(Class 6) Legal	8	11,683
(Class 7) Educational Instruction and Library	68	46,240
(Class 8) Arts, Design, Entertainment, Sports, and Media	45	29,269
(Class 9) Healthcare Practitioners and Technical	96	72,353
(Class 10) Healthcare Support	20	19,070
(Class 11) Protective Service	28	19,843
(Class 12) Food Preparation and Serving Related	18	15,999
(Class 13) Building and Grounds Cleaning and Maintenance	10	25,654
(Class 14) Personal Care and Service	34	15,672
(Class 15) Sales and Related	23	20,483
(Class 16) Office and Administrative Support	55	38,024
(Class 17) Farming, Fishing, and Forestry	14	17,747
(Class 18) Construction and Extraction	65	24,076
(Class 19) Installation, Maintenance, and Repair	52	32,628
(Class 20) Production	114	40,429
(Class 21) Transportation and Material Moving	57	32,892
(Class 22) Military Specific	19	–
Total	**1016**	**692,600**

This data extraction process was carried out in January 2025. For each job posting, the following information was collected: (1) job title, (2) job description, (3) O*NET occupational family code and (4) O*NET occupation code.

If we compare this dataset with those used in previous studies mentioned in Section 2, we observe that both the size and the variety of job positions included in this study are significantly greater. For example, in Skondras et al. [1], two data sets were used: one with 2250 real job postings collected from the Indeed.com portal and another with 4791 synthetic postings generated with ChatGPT. This dataset includes job descriptions and their classification into 15 occupational families. In Maitra et al. [2], a dataset of 137,714 job postings was created from seven open-source resources, classified based on job titles. In Clavié et al. [3], an additional dataset of 10,000 job postings was used to classify whether the positions were suitable for recent graduates. Finally, in Beristain et al. [4], a Kaggle dataset containing 200,000 job postings was employed.

A study was conducted on the length of the job descriptions from the selected postings, as this field was used in the fine-tuning process (see Table 2). The average length of job descriptions is around 635 words, with a range from 1 to 8250 words. The interquartile range indicates that the central 50% of the descriptions contain between 384 words (Q1) and 827 words (Q3), with a median length of 573 words. The difference between the mean and the median suggests an asymmetric, positively skewed distribution, due to the presence of several considerably long texts that increase the overall average. The occupational family with the highest average description length was observed for Protective Service, with 849.54 words, while the occupational family with the lowest average description length is Farming, Fishing and Forestry, with 476.20 words.

Table 2: Descriptive statistics of job descriptions (words)

Total descriptions		Family with longer descriptions		Family with shorter descriptions	
Statistic	**Value**	**Statistic**	**Value**	**Statistic**	**Value**
Count	692,600	Count	19,843	Count	17,747
Mean	635.49	Mean	849.54	Mean	476.20
Std Dev	421.62	Std Dev	668.61	Std Dev	327.25
Min	1	Min	2	Min	2
25%	384.00	25%	499.50	25%	303.00
Median	573.00	Median	665.00	Median	424.00
75%	827.00	75%	998.00	75%	593.50
Max	8250	Max	6906	Max	6727

Fig. 1 shows the histogram of text lengths, where a high concentration of descriptions is observed around the median, together with outliers at both extremes of the distribution.

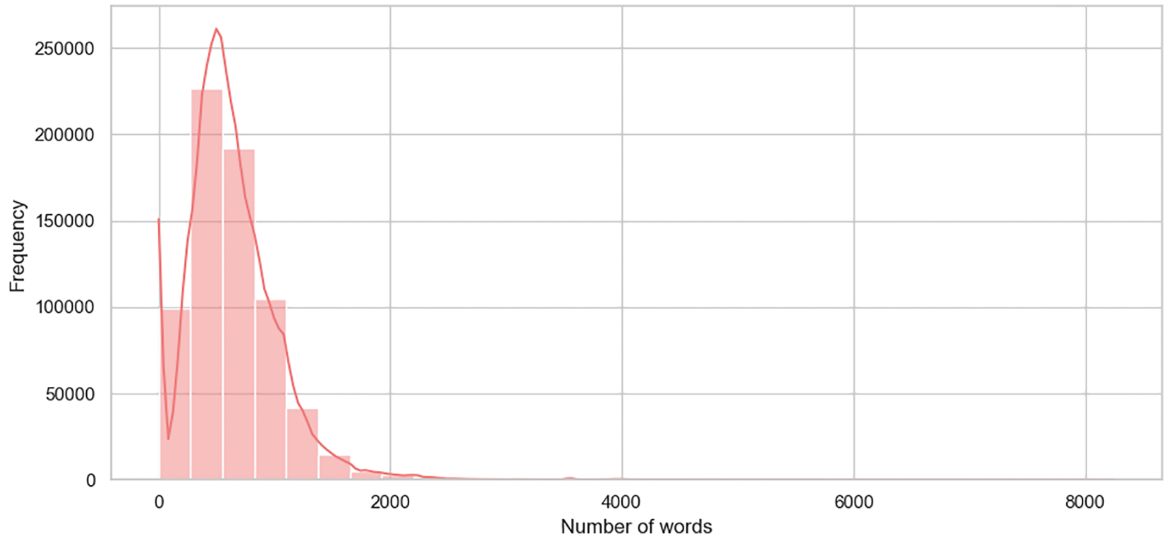

Figure 1: Distribution of words in job descriptions

3.2 Dataset Pre-Processing

First, the dataset was processed to remove existing outliers. When we examined the shortest descriptions, we observed that some contained no meaningful content or consisted of very short texts that did not

define the position; therefore, we removed all job offers whose descriptions had fewer than 50 words. We then analyzed the longest descriptions and found that most contained boilerplate text, legal disclaimers, or organizational information unrelated to the posting, thereby hindering the classification process. These long descriptions were shortened during the subsequent tokenization stage to retain only the initial segment.

Subsequently, the description field was preprocessed using the following pipeline [17]: (1) lower case reduction, (2) URLs removal, (3) HTML tags removal, (4) punctuation removal, and (5) tokenization. Since BERT and its variants, such as RoBERTa and DeBERTa, have a limit in their context windows of 512 tokens, we applied truncation during tokenization to process only the initial portion of each description. This strategy shortens long descriptions while retaining the opening segment, where the most relevant information for classification—such as the job title, requirements, and key competencies—is typically found. Therefore, we expect that these input constraints do not significantly affect the models' ability to capture the semantic patterns that distinguish each category.

This preprocessing aims to preserve the structure and richness of the natural language present in the original data, as it has been shown that transformer-based models, such as BERT, achieve better performance when working with texts that retain their original linguistic characteristics [18].

The output of preprocessing the description field will serve as the input variable, and the O*NET occupational family code will be defined as the target variable. The resulting dataset was randomly split into training and test subsets (80%–20%) [19]. Table 3 presents descriptive statistics of the words and tokens, respectively, in the job descriptions from the training and test datasets after preprocessing.

Table 3: Descriptive statistics of datasets (words and tokens)

Training dataset (words)		Testing dataset (words)		Training dataset (tokens)		Testing dataset (tokens)	
Statistic	**Value**	**Statistic**	**Value**	**Statistic**	**Value**	**Statistic**	**Value**
Count	522,177	Count	130,545	Count	522,177	Count	130,545
Mean	674.21	Mean	673.60	Mean	483.00	Mean	482.61
Median	596.00	Median	598.00	Median	512.00	Median	512.00
Std Dev	403.91	Std Dev	401.05	Std Dev	74.06	Std Dev	74.60
Min	50	Min	50	Min	58	Min	61
Max	8250	Max	8250	Max	512	Max	512

As can be observed, the shortest descriptions have a length of 58 tokens, while the longest ones reach 512 tokens. After tokenization, the average length is 483 tokens, with a median of 512 tokens. This indicates that many descriptions were truncated to the maximum input length, yet they still retain a substantial amount of useful information. It is also worth noting that the standard deviation, minimum, and maximum values are very similar between the training and test datasets. This suggests that no bias was introduced during the data split, and that the models will be exposed to representative examples in both the training and evaluation phases.

Finally, to assess the impact of training with an imbalanced dataset, we created a new balanced dataset using a weighted sampler (WeightedRandomSampler) in PyTorch, which adjusts each instance's sampling probability according to its class to counteract class imbalance. As a result, we obtained a balanced training set with 225,957 records and a test set with 25,107 records.

3.3 Fine Tuning and Evaluation

We have selected three language models that are frequently employed for classification tasks:

- BERT (Pre-Training of Deep Bidirectional Transformers for Language Understanding) [19]: BERT is the first bi-directional (or non-directional) pretrained language model. Through fine-tuning, the model can be used for different classification tasks.
- RoBERTa (A Robustly Optimized BERT Pretraining Approach) [20]: Roberta is an evolution of BERT and achieves better performance in all tasks. Its architecture is similar to BERT with modifications in the hyperparameters. This results in improved performance on various natural language understanding tasks, including job classification. RoBERTa's architecture allows it to better generalize across different job descriptions, making it particularly effective for classifying diverse job offers. Studies have shown that RoBERTa outperforms BERT in many benchmarks, indicating its suitability for tasks requiring high accuracy in classification [20].
- DeBERTa (Decoding-enhanced BERT with disentangled attention) [21]: DeBERTa has a similar architecture and size to BERT but obtains better performances by incorporating a disentangled attention mechanism. This allows DeBERTa to capture more intricate relationships between words, leading to improved performance in understanding complex job descriptions. The model's ability to handle nuanced language makes it particularly effective for classifying job offers that may contain ambiguous or specialized terminology [22].

As mentioned in Section 2.2, previous studies have typically used the BERT model for the task of job offer classification. However, there is no study that compares the three most important models of the BERT family—BERT, RoBERTa, and DeBERTa—for this specific task.

An initial analysis of the training dataset revealed the presence of class imbalance, with class sizes ranging from 11,683 to 72,353 samples. Consequently, the models were initially trained using the entire dataset to avoid information loss and to promote generalization. Subsequently, training will also be performed on a balanced dataset in order to compare the results obtained in each scenario.

Given the size of the dataset and the computational demands of retraining, an initial training phase was performed to evaluate the impact of the number of epochs on the model's performance. For this preliminary analysis, the BERT architecture was employed for text classification, utilizing a set of hyperparameters that have been widely validated in the literature. The learning rate was set to 2e–5, a value that has demonstrated robust and stable performance across similar classification tasks in the GLUE benchmark [19,23]. A batch size of 16 was selected to balance computational efficiency and training stability [23,24]. Furthermore, a weight decay coefficient of 0.01 was applied as L2 regularization, in accordance with the guidelines proposed by Liu [20] to improve generalization. A linear learning rate scheduler with a warm-up period corresponding to 10% of the total training steps was also used, a strategy that has been shown to improve the stability of the training in pre-trained language models [20]. Using this base configuration, three experiments were conducted with 3, 5, and 10 training epochs, employing both the full dataset and the balanced dataset.

Subsequently, a second training phase was conducted employing the three specified models: BERT, RoBERTa, and DeBERTa. Hyperparameter tuning was performed individually for each model using `Optuna` [25], a Bayesian optimization framework designer for the automatic and efficient search of hyperparameters through pruning and sequential search techniques. The search space included the following hyperparameters: learning rate, batch size, and weight decay. Each hyperparameter configuration was evaluated after a complete training cycle of 10 epochs, utilizing the `Trainer` class from the `Transformers` library. To mitigate overfitting, an `EarlyStoppingCallback` with a patience of 2 epochs was implemented, allowing early termination of training if model performance declined.

Finally, an evaluation phase was conducted for the retrained models using the test dataset and standard metrics for multiclass classification tasks, including accuracy, precision, recall, and macro F1-score. These metrics were computed using the `scikit-learn` library [17].

4 Results and Discussion

In the initial training phase, the `bert-base-uncased` model was used while varying the number of epochs (3, 5, and 10). As shown in Table 4, increasing the number of training epochs did not lead to improvements in model performance, and the best metrics were achieved with the training run using 3 epochs (Accuracy = 0.764, Precision = 0.765, Recall = 0.765, macro F1-score = 0.764). This behavior can be attributed to the high capacity of large pretrained language models such as BERT, which typically converge rapidly during fine-tuning, especially when trained on large datasets. In this context, the model can quickly capture the task-relevant patterns and decision boundaries in the early epochs. Continuing training beyond this point often leads to diminishing returns and may even increase the risk of overfitting, negatively affecting the model's generalization to unseen data.

Table 4: Results of retraining the `bert-base-uncased` model using the full dataset

Configuration	Accuracy	Precision	Recall	Macro F1-score
3 epochs	0.764	0.765	0.765	0.764
5 epochs	0.761	0.763	0.761	0.761
10 epochs	0.758	0.759	0.758	0.758

Subsequently, a training phase analogous to the previous one was conducted, this time using the balanced dataset, yielding the results reported in Table 5. The best results were also obtained with 3 training epochs, achieving an accuracy of 0.762 and a macro F1-score of 0.762. As with the full dataset, increasing the number of epochs beyond this point did not improve performance and, in fact, slightly reduced it, likely due to overfitting.

Table 5: Results of retraining the `bert-base-uncased` model using the balanced dataset

Configuration	Accuracy	Precision	Recall	Macro F1-score
3 epochs	0.762	0.762	0.761	0.762
5 epochs	0.761	0.762	0.760	0.760
10 epochs	0.757	0.757	0.756	0.757

The fine-tuning process using the balanced dataset resulted in performance very similar to that achieved when training with the full, imbalanced dataset. The balanced dataset contributed to a fairer representation across classes, potentially improving robustness for minority classes, although it did not lead to a significant improvement in overall metrics. Additionally, it reduced the training time from 3.2 to 2.4 h per epoch, allowing for greater computational efficiency, cost reduction, and the ability to iterate faster in experiments without sacrificing model quality.

Finally, a hyperparameter tuning process was conducted using the balanced dataset and the following models: `bert-base-uncased`, `roberta-base`, and `deberta-base`. The objective function evaluated combinations of learning rates (2e–5, 3e–5, 5e–5), batch sizes (8, 16, 32), and weight decay values (0.01, 0.1) across 18 experiments, using the macro F1-score on the validation set as the primary performance metric.

Each experiment consisted of a full training cycle of 10 epochs with an `EarlyStoppingCallback` parameter set to a patience of 2 epochs. Table 6 shows the best results obtained for each model and the hyperparameter configuration used in that iteration.

Table 6: Results of model retraining and hyperparameter tuning

Model	Configuration	Accuracy	Precision	Recall	Macro F1-score
Bert-base-uncased	Learning rate = 3e–5 Batch size = 8 Weight decay = 0.1 Number of epochs = 3	0.762	0.763	0.763	0.763
Roberta-base	Learning rate = 2e–5 Batch size = 8 Weight decay = 0.01 Number of epochs = 4	0.765	0.766	0.765	0.764
Deberta-base	Learning rate = 2e–5 Batch size = 8 Weight decay = 0.01 Number of epochs = 6	0.768	0.767	0.768	0.767

The results reveal a consistent improvement across models, with DeBERTa outperforming both BERT and RoBERTa in all metrics. This progression aligns with the evolution of model architectures, suggesting that more recent and sophisticated models capture richer contextual representations, leading to better classification performance. It is worth noting that both RoBERTa and DeBERTa required a greater number of training epochs to achieve their best results compared to BERT. This behavior can be attributed to the increased complexity and capacity of these architectures, which demand a longer fine-tuning period to optimally learn the relevant patterns in the dataset.

The best overall performance was achieved by DeBERTa, obtaining an Accuracy of 0.768 and a macro F1-score of 0.767, which is competitive given the complexity of the dataset and the large number of classes. A key strength observed in all models is the balanced performance between Precision and Recall, which results in relatively stable macro F1-scores and indicates that the models do not overly favor one metric at the expense of the other. These results are similar to those obtained in other job offer classification studies; for example, Maitra et al. [2] achieved an F1-score of 0.78 using a BERT model in his classification process.

However, the improvements between models, while consistent, remain moderate. For instance, the macro F1-score increases by only 0.004 from BERT to DeBERTa. This suggests that both the fine-tuning process and the characteristics of the dataset—including possible similarities among the occupational families to be classified—might be limiting the potential for further performance gains. When calculating the 95% confidence interval, we observe that BERT achieved a macro F1-score of 0.763 (0.757–0.768), RoBERTa 0.764 (0.759–0.770), and DeBERTa 0.767 (0.762–0.772). Although DeBERTa showed the highest averages, the confidence intervals overlap; therefore, no statistically significant differences are observed among the results of the three models.

The analysis of the confusion matrix (see Fig. 2) and the per-class metrics for the DeBERTa model (see Table 7) show that the model achieves high accuracy across most classes, as evidenced by the predominant values along the main diagonal, reflecting numerous correct predictions.

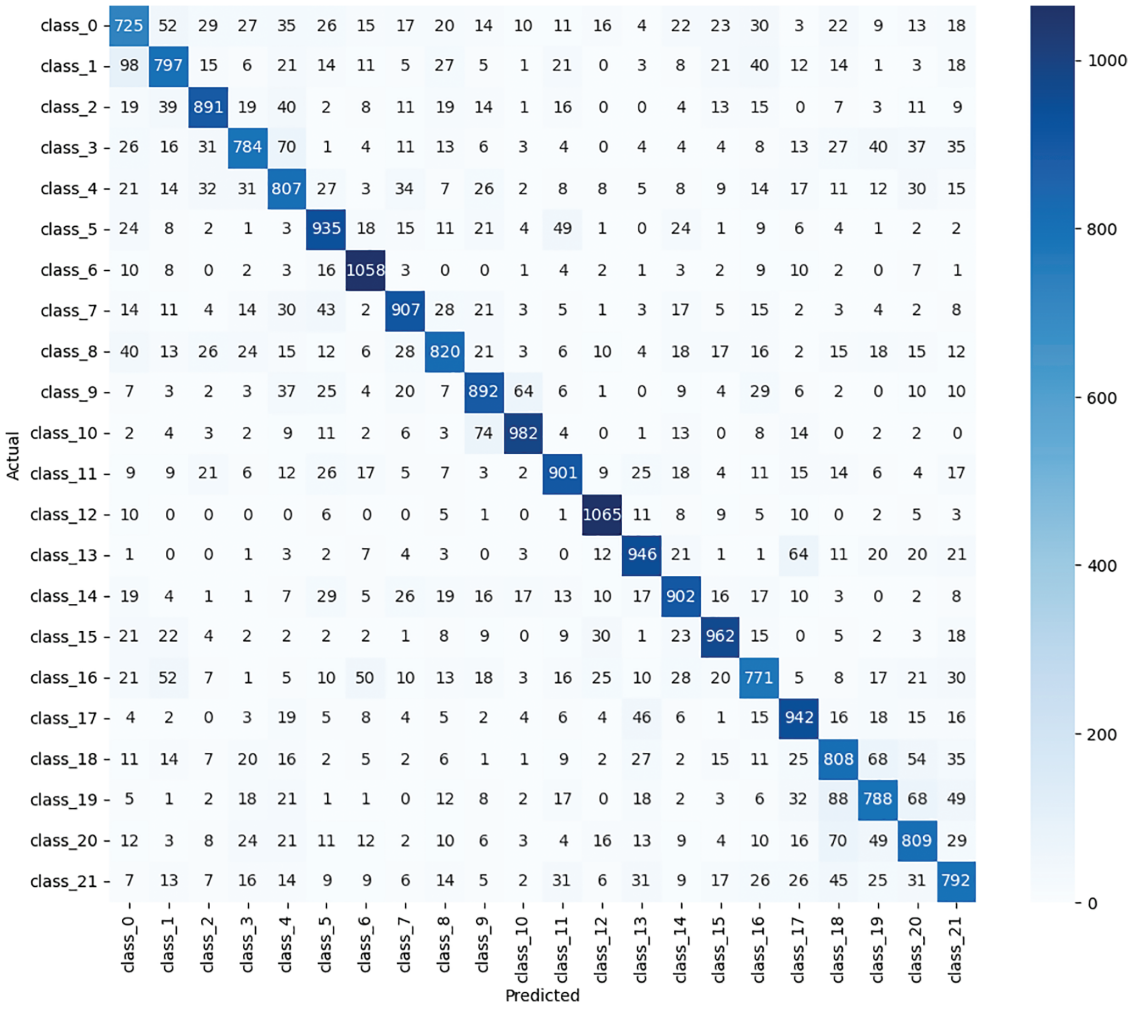

Figure 2: Confusion matrix for the DeBERTa model

Table 7: Per-class metrics for the DeBERTa model

	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7	Class_8	Class_9	Class_10	Class_11	Class_12	Class_13	Class_14	Class_15	Class_16	Class_17	Class_18	Class_19	Class_20	Class_21
Precision	0.655	0.734	0.815	0.780	0.678	0.769	0.848	0.811	0.775	0.766	0.883	0.789	0.874	0.808	0.778	0.835	0.713	0.765	0.687	0.726	0.695	0.691
Recall	0.635	0.698	0.780	0.687	0.707	0.819	0.926	0.794	0.718	0.781	0.859	0.789	0.933	0.829	0.789	0.843	0.675	0.825	0.708	0.690	0.709	0.694
F1-Score	0.645	0.716	0.798	0.730	0.692	0.793	0.885	0.803	0.746	0.774	0.871	0.789	0.902	0.818	0.784	0.839	0.693	0.794	0.697	0.707	0.701	0.692

The best performance is observed in categories with highly distinctive vocabulary and relatively narrow semantic contexts: Food Preparation and Serving Related (Class 12, F1-score = 0.902), Legal (Class 6, F1-score = 0.885), and Healthcare Support (Class 10, F1-score = 0.871). These families typically include technical terminology, certifications, credentials, or very specific tasks that facilitate class discrimination and reduce confusion with other classes.

Mid-range performance is observed in categories such as Sales and Related (Class 15, F1-score = 0.839), Building and Grounds Cleaning and Maintenance (Class 13, F1-score = 0.818), and Educational Instruction

and Library (Class 7, F1-score = 0.803). Although these areas exhibit some lexical overlap with adjacent domains (e.g., sales with marketing or customer service; cleaning/maintenance with concierge or facility management), they maintain sufficiently consistent core terminology and task descriptors to achieve solid performance.

Lower performance generally corresponds to cross-cutting or highly overlapping families: Management (Class 0, F1 = 0.645), Life, Physical, and Social Science (Class 4, F1 = 0.692), Transportation and Material Moving (Class 21, F1 = 0.692), Office and Administrative Support (Class 16, F1 = 0.693), and Construction and Extraction (Class 18, F1 = 0.697). In Management and Business and Financial Operations, job postings often use generic language (e.g., leadership, management, planning, budgeting) shared across multiple domains and hierarchical levels, making boundaries harder to delineate. Office and Administrative Support includes terms common in business/finance or legal roles (e.g., documentation, coordination, compliance), which fosters confusion with offers in Business and Financial Operations and Legal. In the Construction/Installation/Production/Transportation block (Classes 18, 19, 20, 21), common tasks and tools (e.g., maintenance, inspection, safety, logistics) tend to raise cross-class error rates among these families.

These per-class differences can be explained by (i) lexical distinctiveness (classes with clear technical jargon perform better), (ii) semantic ambiguity and cross-cutting roles (e.g., Management/Office/Business), (iii) functional proximity among contiguous families (Construction, Installation, Production, Transportation), (iv) possible label noise and variability in job-posting phrasing (mixed responsibilities, requirements, and corporate templates), and (v) class-wise data distribution (balanced training helps but does not fully eliminate rarity or intra-class diversity effects).

In summary, although the model performs well across most classes, there remains room to reduce systematic errors in specific groups, potentially by leveraging more concrete and domain-specific job descriptions.

5 Conclusions and Outlook

In this study, we performed an exhaustive fine-tuning process on three pretrained state-of-the-art language models (BERT, RoBERTa, and DeBERTa) for a job offers classification task based on O*NET occupational categories. The results demonstrate an improvement across models, with DeBERTa achieving the best performance (macro F1-score = 0.767). This underscores the importance of leveraging newer architectures that more effectively capture contextual information, although the gains between models are relatively modest (the macro F1-score increases by only 0.004 from BERT to DeBERTa) which may not justify the added model complexity and the longer fine-tuning and inference times required.

These results suggest potential limitations related to the presence of short descriptions that do not adequately detail the job position, and longer descriptions that exceed the maximum number of tokens supported by this family of models. Furthermore, semantic overlap among certain categories (e.g., Management, Business and Financial Operations, and Office and Administrative Support) may further constrain the model's classification capacity, leading to significantly lower performance for these categories compared to others that are more distinguishable.

These job offers classification systems are highly useful in recruitment platforms and job portals, as they enable the normalization and structuring of postings according to the O*NET ontology. In HR-specific ATS (Applicant Tracking System) or CRM (Customer Relationship Management) environments, the automatic classification of each new vacancy enables consistent filtering, rule-based routing, comparative analytics across regions and time periods, and better alignment between job postings and candidate profiles. Moreover,

semantic normalization in classification facilitates mapping to other international taxonomies (e.g., SOC or ESCO), improving system interoperability and auditability.

In real-world deployments, the primary benefits include reduced labeling time and cost, increased consistency across departments and companies, fewer human errors and biases, and improved matching between vacancies and CVs when both are classified under the same occupational hierarchy. In high-volume settings, this automation helps absorb demand spikes, sustain labeling quality, and provide additional insights that support workforce planning and training programs.

Analyzing the results obtained with the three models, BERT emerges as the most practical implementation choice: it achieves performance metrics comparable to the other two models while requiring fewer computational resources for both training and inference.

In the experimental environment used in this work (Intel Xeon Gold 6230, 256 GB RAM, and an NVIDIA RTX A5000 GPU), we observe consistent efficiency differences across architectures. For single-inference, BERT requires approximately 14–28 ms, compared with 18–33 ms for DeBERTa; similarly, peak inference memory usage is around 1.2 GB for BERT vs. roughly 1.6 GB for DeBERTa. Consequently, under these conditions, DeBERTa requires about 20%–30% more time and resources per inference.

Looking ahead, future work could explore several directions to improve classification performance. These include expanding the dataset and ensuring better balance among classes, experimenting with advanced data augmentation strategies, using extended-length models (such as Longformer or BigBird), or applying ensemble methods to combine the strengths of different models. Additionally, the use of large language models (LLMs) could be explored to assess whether their advanced capabilities in natural language understanding and generation can lead to improved classification performance [26]. Given the demonstrated success of LLMs in various natural language processing tasks, investigating their application in job offer classification could provide valuable insights and potentially overcome some of the limitations observed in BERT models.

We could also leverage the O*NET hierarchy to perform hierarchical evaluation and classification. For example, job postings could first be assigned to broad occupational categories and then the prediction refined to specific occupations within each category, thereby reducing systematic confusion among adjacent categories (e.g., Construction/Installation/Production/Transportation).

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Lino Gonzalez-Garcia, Miguel-Angel Sicilia, Elena García-Barriocanal; data collection: Lino Gonzalez-Garcia; analysis and interpretation of results: Lino Gonzalez-Garcia, Miguel-Angel Sicilia; draft manuscript preparation: Lino Gonzalez-Garcia, Elena García-Barriocanal. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All code and scripts used for data collection, preprocessing, and model training are publicly available at https://github.com/LinoGonzGar/Classification-job-offers-ONet-BERT-models (accessed on 15 September 2025) [17]. By using the provided scripts, it is possible to generate a dataset comparable to that used in this study and to replicate the methodology described herein.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Skondras P, Zervas P, Tzimas G. Generating synthetic resume data with large language models for enhanced job description classification. Future Internet. 2023;15:363. doi:10.3390/fi15110363.
2. Maitra M, Sinha S, Kierszenowicz T. An improved BERT model for precise job title classification using job descriptions. In: 2024 IEEE 12th International Conference on Intelligent Systems (IS); 2024 Aug 29–31; Golden Sands, Bulgaria. p. 1–6.
3. Clavié B, Ciceu A, Naylor F, Soulié G, Brightwell T. Large language models in the workplace: a case study on prompt engineering for job type classification. In: Métais E, Meziane F, Sugumaran V, Manning W, Reiff-Marganiec S, editors. Natural language processing and information systems. Cham, Switzerland: Springer Nature; 2023. p. 3–17. doi:10.1007/978-3-031-35320-8_1.
4. Beristain SI, Barbosa RRL, Barriocanal EG. Improving jobs-resumes classification: a labor market intelligence approach. Int J Inform Technol Decis Making. 2024;23(4):1509–25.
5. Ruco A, Pinto A, Nisenbaum R, Ho J, Bellicoso E, Hassen N, et al. Collecting occupation and hazards information in primary care using O*NET. Am J Ind Med. 2022;65:783–9. doi:10.1002/ajim.23420.
6. Chiarello F, Fantoni G, Hogarth T, Giordano V, Baltina L, Spada I. Towards ESCO 4.0—is the European classification of skills in line with Industry 4.0? A text mining approach. Technol Forecast Soc Change. 2021;173:121177. doi:10.1016/j.techfore.2021.121177.
7. Gonzalez-Garcia L, Sicilia MA, García-Barriocanal E. Using vector databases for the selection of related occupations: an empirical evaluation using O*NET. Big Data Cognit Comput. 2025;9(7):175. doi:10.3390/bdcc9070175.
8. Dembe AE, Yao X, Wickizer TM, Shoben AB, Dong XS. Using O*NET to estimate the association between work exposures and chronic diseases. Am J Ind Med. 2014;57:1022–31. doi:10.1002/ajim.22342.
9. Gadermann A, Heeringa SG, Stein MB, Ursano RJ, Colpe LJ, Fullerton CS, et al. Classifying u.s. army military occupational specialties using the occupational information network. Military Med. 2014;179:752–61.
10. Babiuk V, Adisesh A, Baker CJO. S-469 An integrated database of occupational information from O*NET-SOC and the canadian career handbook. Occup Environ Med. 2021;78(Suppl 1):A1–A173. doi:10.1136/oem-2021-epi.443.
11. Sreerambatla S. Synthesizing job market data: building a unified repository using ONET and ESCO. Int J Sci Res (IJSR). 2020;9(1):1942–6. doi:10.21275/sr24628183641.
12. Alonso R, Dessi D, Meloni A, Recupero DR. A novel approach for job matching and skill recommendation using transformers and the O*NET database. Big Data Res. 2025;39:100509. doi:10.1016/j.bdr.2025.100509.
13. Reimers N, Gurevych I. Sentence-bert: sentence embeddings using siamese bert-networks. arXiv:1908.10084. 2019.
14. Decorte JJ, Van Hautte J, Demeester T, Develder C. Jobbert: understanding job titles through skills. arXiv:2109.09605. 2021.
15. Xu D, Yang H, Rizoiu MA, Xu G. From occupations to tasks: a new perspective on automatability prediction using BERT. In: 2023 10th International Conference on Behavioural and Social Computing (BESC); 2023 Oct 30–Nov 1; Larnaca, Cyprus. p. 1–7.
16. Alonso R, Dessi D, Meloni A, Murgia M, Recupero DR, Scarpi G. A seamless ChatGPT knowledge plug-in for the labour market. IEEE Access. 2024;12:155821–37. doi:10.1109/access.2024.3485111.
17. González García L. Classification of job offers into job positions using O*NET and BERT language models. GitHub Repository; 2025 [Internet]. [cited 2025 Oct 10]. Available from: https://github.com/LinoGonzGar/Classification-job-offers-ONet-BERT-models.
18. Siino M, Tinnirello I, La Cascia M. Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. Inf Syst. 2024;121:102342. doi:10.1016/j.is.2023.102342.
19. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. 2019.
20. Liu Y. Roberta: a robustly optimized bert pretraining approach. arXiv:1907.11692. 2019.
21. He P, Liu X, Gao J, Chen W. Decoding-enhanced bert with disentangled attention. arXiv:2006.03654. 2020.

22. Skondras P, Zótos N, Lagios D, Zervas P, Giotopoulos KC, Tzimas G. Deep learning approaches for big data-driven metadata extraction in online job postings. Information. 2023;14(11):585. doi:10.3390/info14110585.
23. Sun C, Qiu X, Xu Y, Huang X. How to fine-tune BERT for text classification? arXiv:1905.05583. 2019.
24. Dodge J, Ilharco G, Schwartz R, Farhadi A, Hajishirzi H, Smith NA. Fine-tuning pretrained language models: weight initializations, data orders, and early stopping. arXiv:2002.06305. 2020.
25. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, NY, USA: ACM; 2019. p. 2623–31. doi:10.1145/3292500.3330701.
26. Vajjala S, Shimangaud S. Text classification in the LLM Era—where do we stand? arXiv:2502.11830. 2025.