Classification of Product Manager Types to Understand the Job Market

Lena Corredor[1] **Saliia Asanova**[1]* **Zikai Liu**[1]*
[1]University of California, Berkeley
{lena_corredor, saliia.asanova, zikailiu}@berkeley.edu

1 Introduction

Determining the viability of a career path is a difficult task. This is not only the case for students and recent graduates, but also the case for experienced and talented individuals. Recently, this has been complicated by factors such as the tech recession and a macro trend written by McKinsey (2023): The average lifespan of a company listed on the S&P 500 was 61 years in 1958 and is now 18 years. This trend creates a lot of volatility in the market and thus impacts companies and individuals. Natural Language Processing and LLMs can play a significant role in helping professionals understand the job market. One task that helps professionals is the classification of jobs into different categories. Since we have knowledge of the product manager (PM) role, our goal is to use NLP techniques to classify PM job postings into technology categories. This helps students and professionals plan for their careers. With this understanding, some may choose to up-skill into specific technologies such as Artificial Intelligence product management. Another task is to calculate the distance between the different classified types using embedding, and visualize them in 2D. This helps a working professional understand the relative effort and time it takes to transition between them and how different the types are.

2 Literature Review

Through the applied NLP course, we learned of important considerations when using NLP models for a specific task. Elements that affect the results are: 1) the data the model is trained on, including the type of data and the size of the data, 2) the feature and model selection, and the 3) evaluation process. The task here is to classify a product manager job posting into different categories. We conducted literature review in areas of job classification, taxonomy, prior studies on product management and NLP methods relating to classification and methods we used.

The application of Natural Language Processing (NLP) to job market analysis has gained traction in recent years, with researchers exploring various techniques for classifying job postings into different categories. Skondras et al. (2023) utilized TF-IDF on job descriptions and cosine similarity to measure the differences between postings, effectively capturing nuances within the professional landscape. While their work encompassed diverse roles like cooks and receptionists, our paper focuses specifically on categories within product management. Nasser and Alzaanin (2020) used TF-IDF to extract important words for their job posting classification model, achieving high evaluation metrics including accuracy, recall, precision, and F-measure. Their success with TF-IDF in a similar domain further justifies its inclusion as a baseline feature engineering technique in the study discussed in our paper. The study by Maitra et al. (2024) used BERT for multi-label classification of job titles and achieved an accuracy of 0.842. However, this paper's focus differs from the research we are analyzing because it concentrated on precise job title classification, not categorizing roles by technology. Varelas et al. (2022) also investigated the potential of NLP techniques for online job vacancy classification. Their work reinforces the relevance and timeliness of applying NLP to the evolving job market, further supporting the motivation behind this paper's exploration of PM role classification.

We also learned more about the PM role in the 2013 paper - What are the roles of software product managers by Maglyas et al. (2013). Although this paper is a while back, it does outline the core responsibilities of a PM: 1) identifying features that prove significant value for the customer; 2) work with engineering and other teams to build the product; 3) gain authority and influence for the product.

This is considered as a general product manager and it's categorized as "Other Type of Product Management" in the taxonomy that we created.

While prior research has laid the groundwork for NLP-based job classification, the specific application to Product Management (PM) roles, particularly with a focus on technology categories, presents a more specialized area of study. Existing literature on PM roles often focuses on defining core responsibilities and competencies Maglyas et al. (2013), but a comprehensive taxonomy of PM roles based on technology expertise remains underdeveloped.

The paper analyzed in the sources builds upon previous work by creating a labeled dataset of PM job postings categorized by technology, employing a combination of established NLP techniques like TF-IDF, Logistic Regression, and BERT. While these methods are not novel in themselves, their application to this specific dataset and task contributes to the research on NLP-driven job market analysis. Furthermore, the paper benchmarks its results against the state-of-the-art GPT-4 CoT model, providing insights into the relative difficulty of the task. Finally, the clustering analysis using BERT embeddings offers a visualization of the relationships between different PM job categories, highlighting the distinct nature of technical roles like AI/ML compared to more traditional PM positions.

3 Data Collection and Preparation

We obtained the job posting dataset from Indeed using a web scraper tool with the search term 'product manager' as of October 2024. This search also picks up non-PM related roles, but has the terms product and manager in the title. E.g. Product marketing manager. From the initial corpus of 5,000 job postings extracted from Indeed, we manually annotated a subset of 782 postings to ensure precise identification of legitimate Product Manager positions. Instead of manually filtering non-PM roles during preprocessing, we dedicated one classification category called: Non-PM, which is for any role that is not computer product manager type. To enhance data quality, we implemented a heuristic filter to retain only postings with more than 200 words. We made two assumptions for our method. We assumed the sample taken is representative of the entire population in the United States. Another assumption is the category of product manager roles are distinct of each other. In another words, a human can identify the prominent type when looking at a job description.

We first created the labeling guidelines for 9 categories of product manager by technology. The 9 categories are listed below. Their labeling guidelines are in this spreadsheet due to their length.

- **AI/ML Product Manager**
- **Platform Product Manager**
- **Cloud Product Manager**
- **Other Types of Product Manager**
- **Fintech Product Manager**
- **Data Product Manager**
- **API/Integrations Product Manager**
- **Trust & Safety Product Manager**
- **Non-PM**

To evaluate annotation reliability, we conducted an Inter-Annotator Agreement (IAA) study among three annotators. The analysis was performed on a randomly selected subset of 15 job postings, with each annotator independently classifying the positions into the 9 categories according to our defined taxonomy. With this, we calculated the inter annotator agreement values using Cohen's Kappa. The Cohen's Kappa values we obtained are:

Annotator Pair	**Cohen's Kappa**
Annotator 1 vs 2	0.4599
Annotator 1 vs 3	0.6847
Annotator 2 vs 3	0.4529

Table 1: Cohen's Kappa for Annotator Agreement

Cohen's Kappa Range	**Agreement**
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost Perfect

Table 2: Common Interpretations of Cohen's Kappa

Given the moderate Cohen's kappa values obtained from initial annotations, we aimed to move annotator understanding to substantial. Three annotators discussed the differences in understanding and made changes to the labeling guidelines so that we agree on the new classification.

Following the refinement of annotation guidelines, we classified a total of 782 job postings across nine categories. The resulted distribution revealed "Other type of PM" and "Non-PM" as the two most prevalent categories, comprising 283 (36.2%) and 231 (29.5%) samples respectively. In contrast, Trust & Safety (12 samples) and API (10 samples) emerged as the least represented ones.

We constructed the training dataset by integrating job titles and descriptions, paired with our defined nine categories as classification labels, forming the basis for our supervised learning task.

4 Method

4.1 Feature Engineering & Model Selection

In this study, we explored various features and models, including TF-IDF, logistic regression, and SentenceBERT embeddings (Reimers, 2019). Additionally, we introduced a chain-of-thought (CoT) prompting approach using GPT-4 for comparison and benchmarking.

TF-IDF (Term Frequency-Inverse Document Frequency) Sammut and Webb (2010) weights words based on their frequency within a document (in this case, job descriptions) and rarity across the dataset, enabling identification of role-specific terms (e.g., "learning" for *AI/ML* PMs, "financial" for *Payments* PMs). Our TF-IDF vectorizer implementation extracts 768 features to match sentence-BERT's dimensionality, incorporating unigrams and bigrams while filtering terms based on document frequency (appearing in 0.3-90% of documents) and removing English stop words. This feature representation serves as input for a Logistic Regression classifier.

By maintaining consistent dimensionality (768) across both BERT and TF-IDF representations, we establish a controlled framework for comparative analysis, though the fundamental nature of these embeddings differs substantially. While BERT generated dense vectors where most dimensions carry semantic information, TF-IDF produces sparse representations where non-zero values directly correspond to term importance scores Sammut and Webb (2010). This architectural difference—dense versus sparse embeddings—provides an interesting lens through which to examine their respective strengths in categorizing PM roles. Given TF-IDF's inherent sparsity, we opted against regularization (C=1.0) Brightwood (2024) in the logistic regression classifier to preserve its representation capacity.

To capture deeper contextual nuances, we employed BERT (Bidirectional Encoder Representations from Transformers) Embeddings (BERT-base-uncased) Devlin et al. (2019), a pre-trained transformer model renowned for its ability to generate context-aware embeddings for textual input. In this work, we fine-tuned BERT to encode both job titles and descriptions as a single input sequence for each classification task, leveraging its bidirectional architecture to understand the words in context. The fine-tuned model was then connected to a custom fully connected layer, optimized for multi-class classification. Our experiments showed a modest 2% improvement in performance using the [CLS] token representation over mean pooling, aligning with established research on the effectiveness of [CLS] tokens in capturing hierarchical contextual relationships Devlin et al. (2019).

To adapt BERT for our application and maximize its performance, we configured the model to process input sequences with a maximum token length of 512, a batch size of 16 during training, and an epoch size of 7, following the recommendations for small sample sizes as outlined in Sun et al. (2019). During training, we implemented early stopping to preserve the model weights that achieved the highest validation accuracy. This choice balances GPU memory efficiency with the ability to capture meaningful long-term dependencies in job descriptions. However, as many job descriptions exceeded the 512-token limit, truncation of text inevitably occurred, which might have led to the loss of critical information. To address this, we tested two preprocessing techniques: a head-only BERT-Base structure to focus on the title and introductory details, and a dynamic regular expression match model BERT-RegEx to extract the job responsibilities section. Additionally, a dropout layer, inspired by its implementation in Sun et al. (2019), was added during fine-tuning to mitigate overfitting. The effectiveness of these configurations in product management role categorization is empirically evaluated in Section 5.

In addition to the TF-IDF and BERT approaches, we compared our models with the state-of-the-art (SOTA) Large Language Model (LLM) with a chain-of-thought (CoT) prompt, specifically employing the GPT-4 CoT Wei et al. (2023). By incorporating an advanced LLM into our experiment, we aimed to benchmark our BERT approaches against a SOTA model thereby gain insights into the relative difficulty of the task. Due to computational constraints, however, we limited the GPT-4 CoT evaluation to a curated subset of 211 samples.

To ensure a robust evaluation of our model's performance, we incorporated two complementary validation techniques: detailed misclassification analysis and bootstrap confidence interval estimation. Misclassification analysis aims to identify instances where the model's predictions deviate from the true labels, providing qualitative insights into common error patterns. To quantitatively assess model performance with statistical rigor, we utilized bootstrap confidence interval (CI) estimation Bestgen (2022) , which involves repeatedly resampling the validation set with replacement (1,000 iterations) to generate a distribution of accuracy scores. From this distribution, we derived the 95% confidence interval for the model's accuracy (Figure 1). Together, these techniques facilitate a deeper understanding of model behavior and aiding in the comparison of different model configurations.

Further discussions on model limitations, particularly concerning token truncation and category imbalance, are presented in Section 5. Our findings emphasize the importance of tailoring pre-trained transformer models for domain-specific tasks.

We employed stratified sampling to split the data into 80% training and 20% testing sets, ensuring that the class distribution was preserved across both subsets. We utilized a single train-test split supplemented with bootstrap confidence intervals to maintain consistent assessment protocols across all models. Four models were chosen for evaluation:

- **Majority Class Baseline**: This baseline model consistently predicts the most frequent class, the category 'Other type of PM', in the data set which comprises 36.91% of the data.

- **Logistic Regression with TF-IDF (LG+TF-IDF)** : This approach achieved a training accuracy of 71.36%, but the validation accuracy decreased to 52.23% (95% Confidence interval: [44.59%, 59.87%]).

- **BERT-base Classifier with Initial 512 Tokens**: Utilizing the first 512 tokens (including both the title and truncated job description), this BERT-Based model attained a training accuracy of 85.81% and a validation accuracy of 61.78% (95% CI: [54.78%, 68.81%]).

- **BERT-RegEx Classifier Enhanced with Regular Expression**: This variant incorporates a regex-based extraction of the "About the Job" section before feeding the data into the BERT model. It achieved both the highest training accuracy 89.92%, and the highest validation accuracy of 64.97% (95% CI: [57.32%, 71.97%]).

- **GPT-4 Chain-of-Thought (CoT) Model**: The SOTA large language model with chain-of-thought reasoning achieved a classification accuracy of 67.30% on a curated subset of 211 samples.

4.2 Clustering

Given BERT's demonstrated effectiveness in capturing PM role semantics through our classification results, we leveraged the base BERT model (bert-base-uncased) without fine-tuning to analyze relationships between job categories. For each posting, we obtained embeddings from the [CLS] token using the pre-trained BERT model, concatenating the job title with RegEx-filtered description text (truncated to 512 tokens). Category-level representations were then constructed by averaging these embeddings within each job type (e.g., AI/ML, Cloud, Platform). To facilitate analysis, we reduced the 768-dimensional embeddings to 2D using PCA Maćkiewicz and Ratajczak (1993) for visualization, while maintaining the original dimensionality for computing pairwise cosine distances between categories.

5 Results

5.1 Classification Results

In this study, we employed three distinct models to classify Product Management (PM) job postings into nine predefined categories based on technology requirements: BERT-Based, BERT-RegEx (utilizing regex for token matching), and TF-IDF + Logistic Regression. Additionally, we benchmarked these models against a GPT-4 CoT (Chain-of-Thought) model. Classification performance was evaluated using accuracy, precision, recall, and F1-score metrics.

The BERT-Based model achieved an overall accuracy of 61.78%, representing a significant improvement over the LG+TF-IDF model's 52.23% accuracy. Specifically, BERT models demonstrated superior performance in categories such as *Cloud*

(F1-score: 0.667 vs. 0.182) and *Payments* (F1-score: 0.588 vs. 0.250), highlighting their ability to better generalize to clearly defined and distinct classes. However, the improvement in accuracy is not statistically significant when considering confidence interval overlap, even with the BERT-RegEx.

The BERT-RegEx model, which integrates regex-based token matching to refine feature extraction, exhibited an overall accuracy of 64.97%, outperforming the BERT-Based model by approximately 3.19%. Notably, the BERT-RegEx model showed substantial category-wise improvements in *AI/ML* (F1-score: 0.800 vs. 0.700) and *Payments* (F1-score: 0.727 vs. 0.588) compared to BERT-Based. These results suggest that for technical domains such as AI and financial services, there is specific area of the job posting that captures the the essence of the job category.

Both BERT variants and TF-IDF model exhibit signs of overfitting, as evidenced by the diverging training and validation loss curves after epoch 3 (Figure 3) and the substantial disparity between training and validation accuracy in TF-IDF.

A notable observation across all models is the zero F1-score for the *API* and *Trust & Safety* categories in Figure 2, indicating a complete failure in identifying these classes. Several hypotheses may explain this. First, number of job postings available for these classes is extremely low (only 2 instances each in validation), which likely leads to inadequate learning during training. In addition, the definitions of API may overlap with other categories or lack distinct linguistic markers, as seen in the LG+TF-IDF's feature weights for each category in Table 3, making it challenging for models to differentiate them accurately.

The BERT-RegEx model achieves a validation accuracy of 64.97%, approaching the performance of our SOTA GPT-4 CoT benchmark (67.30%) for this classification task.

To further understand the performance across models, we conducted a qualitative analysis by examining sample misclassifications for each method. For all models, common misclassification patterns included predicting *Other Type of PM* as *Non-PM* and vice versa (Table 4). This error may arise from the presence of marketing-related (*Non-PM*) project manager roles within the dataset, where job descriptions heavily utilize common product manager keywords. This overlap, although being accurately identified by human labelers, adversely affects model performance. Effectively mimicking human "attention" mechanisms within language models remains a key area for future work to enhance classification accuracy.

Class	**Top Features**
AI/ML	technologies, talent, systems, content, consumer, platform, experiences, world, learning, ai
API	location, development, software development, usa, benefits, agile, partners, partner, digital, google
Cloud	architecture, network, background, microsoft, technical, aws, security, infrastructure, google, cloud
Data	owner, product owner, problems, roadmap, product vision, reporting, capital, aws, analytics, data
Non-PM	issues, brand, manufacturing, required, quality, project, support, product marketing, sales, marketing
Other Type of PM	people, product owner, owner, ad, market, ideas, lines, mobile, digital, product manager
Payments	services, small, businesses, risk, capital, banking, financial, credit, bank, payments
Platform	gather, engineering, applications, risk, mission, features, infrastructure, technical, enterprise, platform
Trust & Safety	*NA*

Table 3: TF-IDF Feature Contribution

We also tested all three methods along with GPT-4 CoT benchmark (Table 5) with a sample posting: Gemini Cloud Assist Senior Product Manager at Google responsible for developing the Gemini (Generative AI) Assist. Human labelers agreed as *AI/ML* by identifying critical terms such as “Gemini” and “Generative AI” within job descriptions. In contrast, simple models like LG+TF-IDF and BERT-Based (with a 512-token limit) misclassify these roles as *Cloud* due to the frequent occurrence of the term “Cloud” in the job posting titled “Gemini Cloud Assist Senior Product Manager, Google Cloud,”. However, the BERT-RegEx model, utilizing regex to extract the “About the Job” section, successfully captured the essential responsibilities, such as “Develop a Gemini Cloud Assist,” and correctly classified the role as *AI/ML*. Similarly, the GPT-4 CoT model employed chain-of-thought reasoning to discern that despite the cloud-related terminology, the primary focus on AI/ML technologies justified the correct classification. This demonstrates that incorporating domain-specific feature extraction and advanced reasoning mechanisms significantly enhances model performance in accurately classifying nuanced job roles.

Model	Top Misclassification Patterns
BERT-Based	1. Platform → Other type of PM (11) 2. Other type of PM → Non-PM(10) 3. Non-PM → Other type of PM (10)
BERT-RegEx	1. Non-PM → Other type of PM (18) 2. Platform → Other type of PM (6) 3. Other type of PM → Non-PM (3)
LG+TF-IDF	1. Non-PM → Other type of PM (14) 2. Other type of PM → Non-PM (13) 3. Platform → Other type of PM (10)

Table 4: Top 3 misclassification patterns for each model (True → Predicted). Numbers in parentheses indicate the count of misclassified instances in validation

Future work should focus on addressing data imbalance through augmentation strategies, refining *API* and *Trust & Safety* class definitions, and exploring hybrid models combining rule-based and transformer architectures. Larger batch sizes and increased regularization techniques could mitigate the observed overfitting issues.

Model	Rationale and Job Snippet
BERT-Based (Cloud)	Focused on the keyword "Cloud" present in the job title: *"Gemini Cloud Assist Senior Product Manager, Google Cloud"*
BERT-RegEx (AI/ML)	Utilized regex to extract job responsibilities: *"Develop a Gemini Cloud Assist"*
TF-IDF (Cloud)	Misled by frequent "Cloud" term without capturing context-specific keywords like "Generative AI"
GPT-4-CoT (AI/ML)	Applied chain-of-thought reasoning to identify key terms: *"Generative AI or Large Language Models"*

Table 5: Comparative Qualitative Analysis of Model Classifications for an AI/ML Role

5.2 Clustering Results

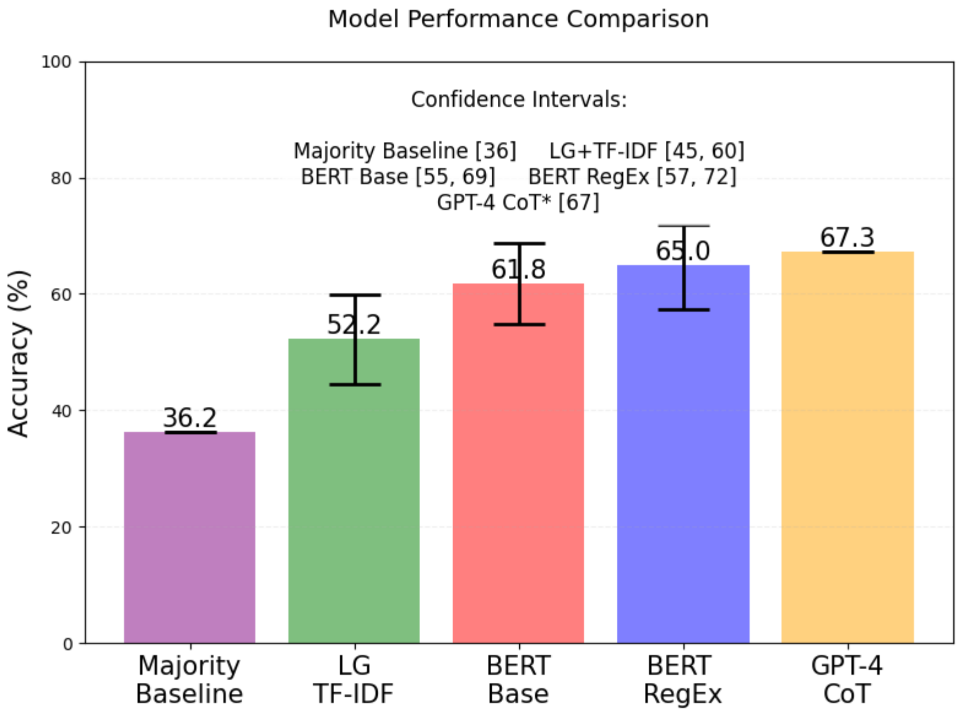
The PCA dimensionality reduction effectively visualizes the embedding space in 2D, capturing 62.90% of the total variance despite some information loss. Our clustering analysis in Figure 4 and Table 6 reveals that technical PM roles show notable differences, with *AI/ML* positions being particularly distinct from other categories (cosine distance of 0.013-0.015 from the *non-PM* roles). This distinctiveness aligns with our model's superior classification performance for *AI/ML* positions. In contrast, traditional PM roles show high semantic similarity, as evidenced by the minimal distances between *Other type of PM*, *Data*, and *Platform* categories (cosine distances of 0.002-0.004). These findings suggest that while core PM competencies remain consistent across most domains, *AI/ML* product management has evolved into a specialized field with unique requirements. For job seekers, this implies that transitioning between traditional PM roles may require less role-specific training compared to moving into *AI/ML* product management.

6 Conclusion

In this study, we developed and evaluated several approaches for classifying product manager job postings across nine technology categories, with our BERT-RegEx model achieving comparable performance to the GPT-4 CoT benchmark. Our analysis revealed that while traditional PM roles share substantial semantic overlap, AI/ML product management positions exhibit distinct characteristics, suggesting specialized skill requirements for these roles. While our models showed promising results for well-defined categories like AI/ML and Cloud, they struggled with under-represented categories such as API and Trust & Safety, highlighting opportunities for future improvements through data augmentation and refined class definitions. These findings provide valuable insights for both job seekers planning career transitions and organizations designing PM role specifications.

Category Pair	Distance
Most Dissimilar Pairs	
Non-PM vs Cloud	0.017
Cloud vs Payments	0.015
Non-PM vs API	0.015
Non-PM vs AI/ML	0.013
AI/ML vs Payments	0.012
Most Similar Pairs	
Other type of PM vs Data	0.002
Non-PM vs Other type of PM	0.002
Other type of PM vs Platform	0.004
Platform vs Data	0.004
Other type of PM vs Payments	0.004

Table 6: Pairwise Cosine Distances Between Category BERT Embeddings



* GPT-4 CoT results are based on 211 selected samples

Figure 1: Model Results

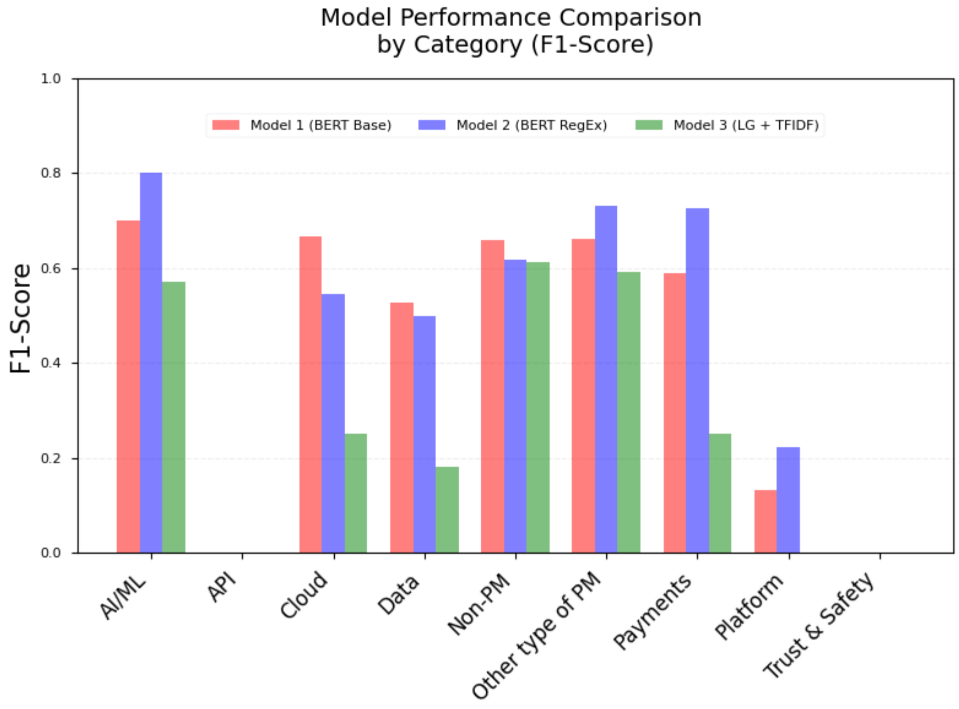


Figure 2: Model Results by Category

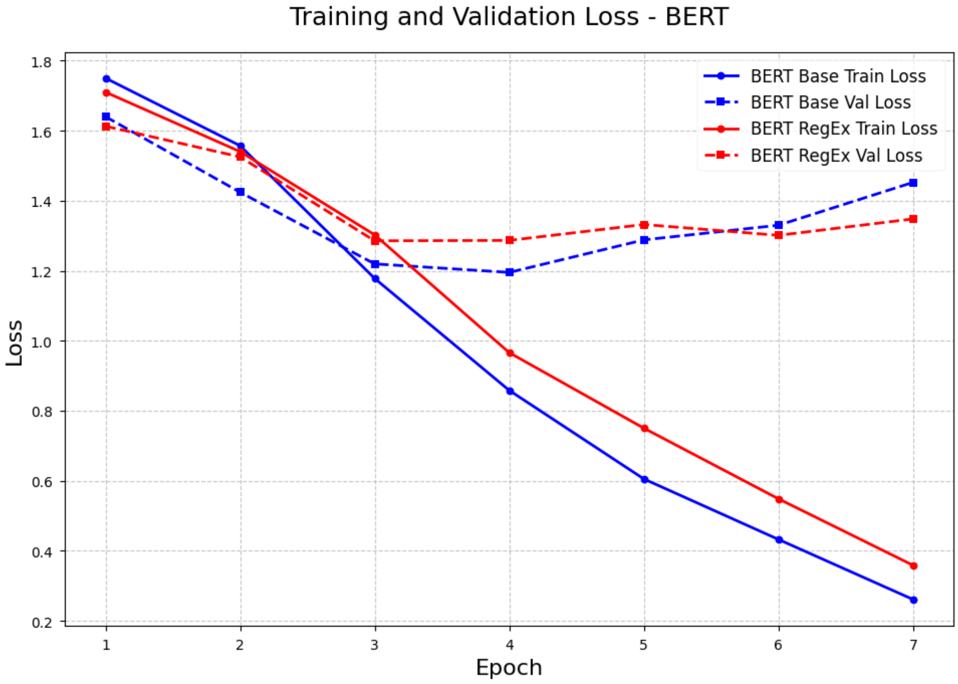


Figure 3: BERT Models Loss Curve

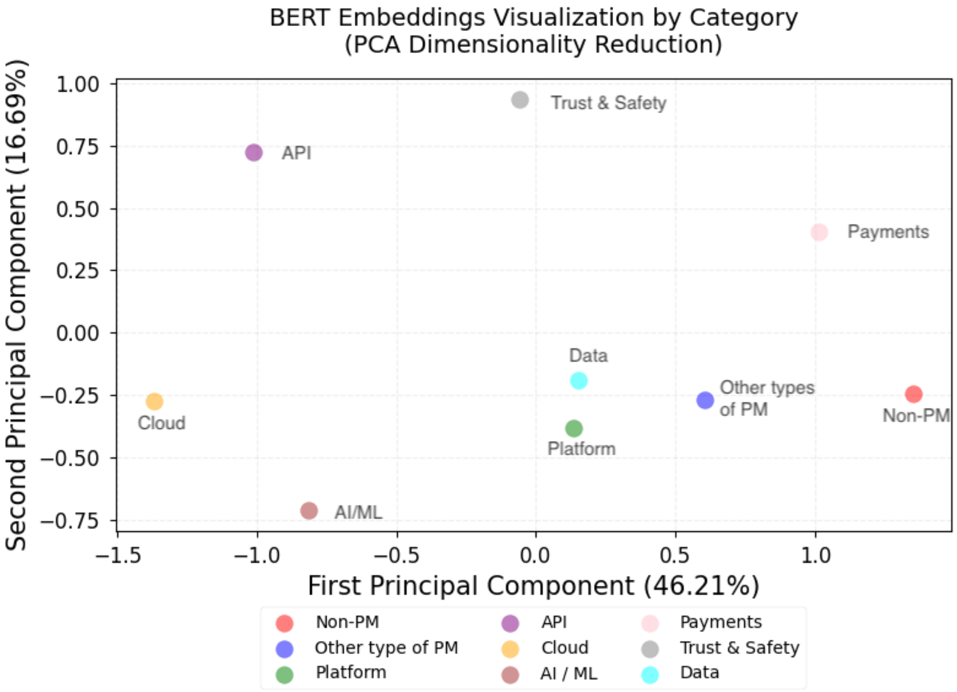


Figure 4: Embeddings by Category in 2D

References

Yves Bestgen. 2022. Please, don't forget the difference and the confidence interval when seeking for the state-of-the-art status. *arXiv preprint arXiv:2205.11134*.

Seraphina Brightwood. 2024. Regularization techniques in logistic regression.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Andrey Maglyas, Uolevi Nikula, and Kari Smolander. 2013. What are the roles of software product managers? an empirical investigation. *Journal of Systems and Software*, 86(12):3071–3090.

Mayukh Maitra, Surabhi Sinha, and Tomas Kierszenowicz. 2024. An improved bert model for precise job title classification using job descriptions. *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, pages 1–6.

Andrzej Maćkiewicz and Waldemar Ratajczak. 1993. Principal components analysis (pca). *Computers Geosciences*, 19(3):303–342.

McKinsey. 2023. Mckinsey on finance. *McKinsey on Finance*. Available online at McKinsey.com.

Ibrahim Nasser and Amjad H. Alzaanin. 2020. Machine learning and job posting classification: A comparative study. *International Journal of Engineering and Information Systems (IJEAIS)*, 4(9):6–14. Available at SSRN: `https://ssrn.com/abstract=3723037`.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Claude Sammut and Geoffrey I. Webb. 2010. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Panagiotis Skondras, Nikos Zotos, Dimitris Lagios, Panagiotis Zervas, Konstantinos C. Giotopoulos, and Giannis Tzimas. 2023. Deep learning approaches for big data-driven metadata extraction in online job postings. *Information*, 14(11).

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? *CoRR*, abs/1905.05583.

George Varelas, Dimitris Lagios, Spyros Ntouroukis, Panagiotis Zervas, Kenia Parsons, and Giannis Tzimas. 2022. Employing natural language processing techniques for online job vacancies classification. In *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops*, pages 333–344, Cham. Springer International Publishing.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.