NLP Predictive Model of Salary from LinkedIn Job Postings

Krysten Nguyen Oona Zhou Ethan Cheng Mansheel Randhawa

1. Introduction

For this project, we aim to develop a classification and forecasting model that utilizes job posting information from Linkedin to extract data and predict salary ranges for specific job titles. The salary information is sometimes absent in the job posting. However, this is crucial knowledge for job seekers, and the lack of information can mislead job seekers and put them in disadvantageous positions. Inspired by this observation, we decided to analyze salary trends of jobs on Linkedin and design a salary forecasting tool using Machine Learning techniques. This classification model can predict salary ranges based on linguistic information including location, job level, job title, company, educational background found in the job description alone.

To approach this question, we compared the performance of different methods to vectorize and embed job descriptions including simple TFIDF, and transformers such as BERT and MPNet. With these vectorizations or embeddings as features and salary ranges as labels, we built supervised classification models including logistic regression, Random Forest, XGBoost, and artificial neural networks (Feed Forward Network, and Long short-term memory).

2. Related Work

We researched 5 scientific paper references for our project.

Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova introduces a new model, BERT. This new model improves on earlier pre-trained unidirectional or shallow bidirectional models. BERT is able to generate deep bidirectional representations by conditioning on both left and right context across all layers. BERT's ability to integrate context from both directions allows it to be fine-tuned to adapt to a broad range of tasks without significant modifications. This flexibility is demonstrated through its performance across multiple NLP benchmarks including GLUE, MultiNLI, and SQuAD, significantly outperforming existing models.

MPNet: Masked and Permuted Pre-training for Language Understanding by Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu and Tie-Yan Liu is the original paper that introduces the pre-training method MPNet in 2020. MPNet combines several models - BERT and XLNet - by leveraging their advantages and handling their limitations. MPNet emphasizes on utilizing the relationships between tokens and recording their position information as an input to ensure accuracy. When pre-trained on a large dataset, MPNet outperforms other models like BERT, XLNet, and RoBERTa on several language understanding tasks. This paper discusses the methodology of MPNet and compares it with other models, as well as introducing the set up of the program.

OversampledML at SemEval-2022 Task 8: When multilingual news similarity met Zero-shot approaches by Mayank Jobanputra and Lorena Martín Rodríguez explores the similarities across multilingual news articles using pre-trained models without any fine-tuning. To analyze the titles and news content, the system applied 14 different similarity features using a combination of pre-trained MPNet with well-known statistical methods, including TF-IDF. The goal is to treat the multilingual news similarity as a regression test and approximate the overall similarity between any two articles. The method achieves a high correlation score, and the strengths and limitations of different features are further analyzed in the paper.

UNLPS at TextGraphs-16 Natural Language Premise Selection Task: Unsupervised Natural Language Premise Selection in Mathematical Text using Sentence-MPNet by Paul Trust, Rosane Minghim, Provia Kadusabe, Ahmed Zahran, Haseeb Younis, and Evangelos Millos utilize a variant of MPNet, Sentence-MPNet, leveraging the capabilities of the model to obtain high-quality sentence embeddings. The Sentence-MPNet modification allows the system to efficiently encode both the premises and the associated mathematical proofs into contextualized embeddings. These embeddings are then used to compute cosine similarities to determine the relevance of each premise to a given proof. This model stands out by improving upon the models that use simpler TF-IDF techniques. This demonstrates the potential of deep learning models like MPNet.

JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks by Nidhi Goyal, Jushaan Singh Kalra, Charu Sharma, Raghava Mutharaju, Niharika Sachdeva, and Ponnurangam Kumaraguru evaluates the performance of the BERT and LSTM models by predicting job skills from descriptions using extreme multi-label classification. LSTMs are favored for their sequence processing capabilities, but often fall short in handling long-term dependencies. On the other hand, BERT is able to better grasp the full context of words across large texts, making it significantly more suitable for the nuanced understanding required to process complex job descriptions. BERT's ability to integrate broader context is particularly emphasized, contrasting with LSTM's limitations in depth of context understanding. The paper suggests that due to the precision of understanding required, BERT's capabilities render it more effective than LSTM.

3. Experiments and Results

Data Preprocessing: The dataset, obtained in CSV format, consisted of 33,246 rows and 28 columns. Initial exploration reveal a subset of 13,352 rows containing salary information in the 'pay_period' column, which served as labels for our classification model. Annual salaries were computed based on median salary and pay period information, facilitating a standardized metric for salary analysis. Salary ranges were categorized into four brackets: 0-50k, 50k-100k, 100k-150k, 150k+. Job description data underwent tokenization, removal of punctuations, emojis, and stopwords to extract meaningful insights. Stemming and stopwords removal techniques were applied to preprocess text data effectively.

Three models were built to get the best-performing classification model. These models were essentially the combination of vectorization/embedding techniques and machine learning or deep learning models.
Model 1: TF-IDF Vectorization + Classification Models (Logistic Regression/ Random Forest/ XGBoost/ Simple Neural Network)
Model 2: Neural Network with Pre-trained BERT
Model 3: Pre-trained MPNet Embedding + Classification Models (Logistic Regression/ Random Forest/ XGBoost/ FNN/ LSTM)

3.1. TF-IDF Vectorization

TF-IDF vectorization was applied to convert text data into numerical vectors. These vectors were then used as input features for various classification models, including Logistic Regression, Random Forest, XGBoost, and a Simple Neural Network.

Logistic Regression: The first machine learning model applied to the TF-IDF vectors was logistic regression. We applied hyperparameter tuning by varying the regularization parameter, C, using the validation set. Different C values (0.01, 0.1, 1, 10, 100) were tested to find the one that maximizes accuracy on the validation set. We then evaluated the model on the test set and calculated the predictions and the probabilities for each class. The F1 score achieved from this model is 0.702 and the AUC score is 0.884, which indicates that the logistic regression model trained on the TF-IDF vectors performed reasonably well in terms of classification accuracy.

SVM: After applying TF-IDF for information retrieval and observing the performance of specific words in the document, we used two machine learning algorithms Random Forest and XGBoost for the task of salary prediction. We trained both models on the filtered dataset and evaluated their performances with the metrics of an accuracy of 0.686, F1 score of 0.684, and an AUC score of 0.874.

Random Forest: For the Random Forest Classifier, we set the model parameters to n_estimators as 1000, max_depth as 50, and bootstrap as True. We used 1000 decision trees with 50 max_depth in the forest, and used bootstrap samples while building the trees. This model has a better performance with an accuracy of 0.6333, F1 Score of 0.6128, and an AUC Score of 0.8877.
The performance indicates that the model successfully classified 63.33% instances, with a harmonic mean of precision and recall and 61.28%, and the ability to distinguish classes is 88.77% as shown in AUC score.

XGBoost: In the XGBoost Classifier, the standard model parameters were set to n_estimators as 1000, and learning_rate as 0.1. We used 1000 decision trees and 0.1 learning rate to prevent overfitting by making the model more robust. This model has a better performance compared to Random Forest, with visible increases of about 10% in accuracy and F1 score, and 3% in AUC Score. We received an accuracy of 0.7536, F1 Score of 0.7517, and an AUC Score of 0.9175. Overall, the XGBoost model outperforms Random Forest classifiers. Therefore XGBoost is better at predicting salary with given job description information.

	Accuracy	F1	AUC
Logistic Regression	0.6985	0.7020	0.8838
SVM	0.6858	0.6940	0.8838
Random Forest	0.6333	0.6128	0.8877
XGBoost	0.7536	0.7517	0.9175

Table 1: Classification Models Using TF-IDF Vectors

3.2. MPNet Embedding

MPNet, a transformer-based model incorporating Masked Language Modeling (MLM) and Permuted Language Modeling (PLM), was used to generate embeddings for job descriptions. These embeddings were then fed into various classification models, including Logistic Regression, Random Forest, XGBoost, Feed Forward Network (FNN), and Long Short-Term Memory (LSTM).

Logistic Regression

We used logistic regression, with a fixed regularization parameter, C, set to 1, was trained and evaluated on the test set by generating predictions and predicting probabilities for each class. The model achieved an accuracy of 0.591. The F1 score was calculated as 0.588 and the AUROC was 0.831. The lower accuracy and F1 score suggest that the model may not perform as well in terms of overall correctness. However, the higher AUROC score shows that the model is good at differentiating between different classes.

Random Forest: First, we created a Random Forest Classifier model with 100 trees and a maximum depth of 20. We trained the model and made predictions from the training data. This model achieved the following results, which is equivalent to the Random Forest model with TF-IDF, without any significant improvement. We received an accuracy of 0.6232, F1 Score of 0.6154, and an AUC Score of 0.8576.

XGBoost: We also tried the XGBoost Classifier with 100 trees, a maximum depth of 20 and 5 output classes. Similarly, after training the model and making predictions based on training data. However, the results were not impressive with significant decreases from the performance of XGBoost Classifier in TF-IDF. On the other hand, its results were similar to the Random Forest Classifier from TF-IDF, with an approximate 2% increase on the F1-score. We received an accuracy of 0.6333, F1 Score of 0.6307, and an AUC Score of 0.8664. MPNet did not achieve the ideal improvement on the model evaluation as we expected, and performed similarly with TF-IDF. Therefore, we concluded that MPNet might not be the most suitable model with its average results.

Feed Forward Network: To further test more models against our data, we tried using a Feed Forward Network for predicting salaries based on job descriptions. A Feed Forward Network is the simplest type of artificial neural network architecture and is widely used in machine learning tasks. We configured this model with two hidden layers, 100 and 50 neurons respectively and utilized an adaptive learning rate with a max of 1000 iterations. Our model performed well with an accuracy of 0.601, F1 Score of 0.602, and an AUC Score of 0.806. These scores indicate that the model was able to classify about 60.11% of the job descriptions correctly, and with a F1 Score close to the accuracy, it suggests a balanced precision and recall. A score of 60.11% reflects that the model learned to predict salary ranges to a reasonable extent. ROC AUC scored 80.59% which suggests the model has good discriminative ability. However, the accuracy achieved is lower than the boosted tree-based model we used such as XGBoost. Feed Forward Network helps us provide a baseline for salary prediction from job descriptions with a simpler model compared to more complex architectures like BERT or LSTM.

LSTM: With a hidden dimension of 100 and an output dimension of 4 representing salary ranges, our LSTM model comprises two layers. The optimizer employed is Adam, with a learning rate set to 0.001. During training, conducted over 10 epochs, we observed a progressive reduction in Cross Entropy loss and improvement in accuracy. Notably, the initial epochs exhibited higher loss values that gradually decreased, with variability in loss values across epochs indicating differing levels of learning difficulty from various job descriptions. On testing, the model yielded the accuracy of around 60%, F1 score of 0.6 and AUROC of 0.836.

The LSTM model, utilizing MPNet embeddings for vectorizing job descriptions, was able to learn the underlying patterns in the data, as evidenced by the decrease in loss and increase in accuracy. While the model has achieved a reasonable baseline performance, XGBoost achieves a marginally better accuracy.

	Accuracy	F1	AUROC
Logistic Regression	0.5914	0.5879	0.8311
Random Forest	0.6232	0.6154	0.8576
XGBoost	0.6333	0.6307	0.8664
Feed Forward Network	0.6011	0.6018	0.8059
LSTM	0.5996	0.5966	0.8356

Table 2: Classification Models Using MP-Net Embeddings

3.3. BERT Embedding

This model leveraged the Bidirectional Encoder Representations from Transformers (BERT) architecture, a state-of-the-art deep learning model for natural language processing tasks.

The BERT model was pretrained on a large corpus of text data and fine-tuned on the job description dataset using a deep learning network. Initialized with pre-trained weights from the 'bert-base-uncased' model, BERT provides contextualized embeddings for input text sequences. Following the BERT model, a dropout layer with a dropout probability of 0.3 is introduced to address overfitting risks. Subsequently, a linear layer with an input dimension of 768 (corresponding to the output dimension of BERT embeddings) and an output dimension of 6 is applied to classify the salary ranges.

The BERT model achieved an accuracy score of 62.21% and F1 scores of 0.6221 (Micro) and 0.6172 (Macro), indicating moderate performance in predicting salary ranges from job descriptions.

The Cross Entropy loss values fluctuated significantly during the training phase, suggesting that the model encountered varying levels of difficulty in learning from different batches. Loss values ranged approximately between 0.5 and 1.7.

4. Discussion

The results show that different machine learning models and NLP techniques can be effective for predicting salaries based on job descriptions. XGBoost consistently outperformed other models, including Random Forest, logistic regression, and MPNet, indicating its suitability for this task. The use of TF-IDF for feature extraction also proved to be effective, especially when combined with XGBoost.

The Feed Forward Network, although simpler compared to other models, provided a reasonable baseline for salary prediction. LSTM, with its ability to capture long-term dependencies, showed some improvement in accuracy but did not outperform XGBoost.

BERT, despite its effectiveness in various NLP tasks, did not perform as well in this study, suggesting that its architecture might not be suitable for capturing the nuances of job descriptions related to salary prediction.

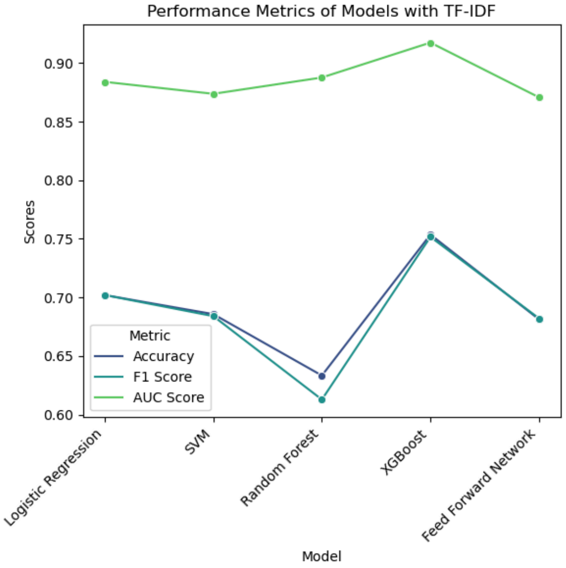

Figure 1: Performance Metrics of Machine Learning Models trained with TF-IDF vectorization

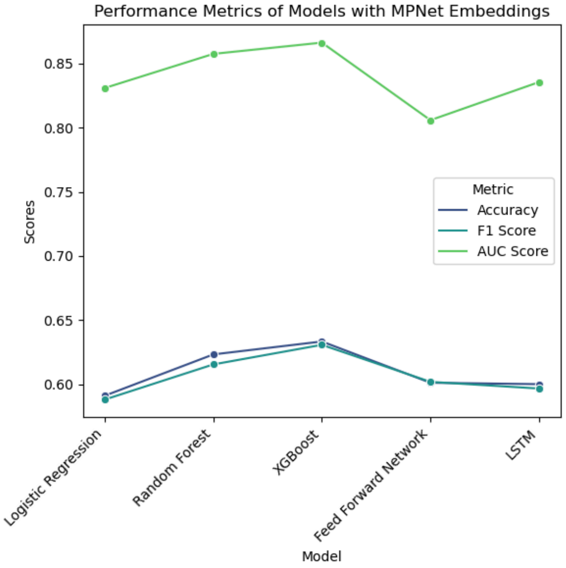

Figure 2: Performance Metrics of Machine Learning Models trained with MPNet Embeddings

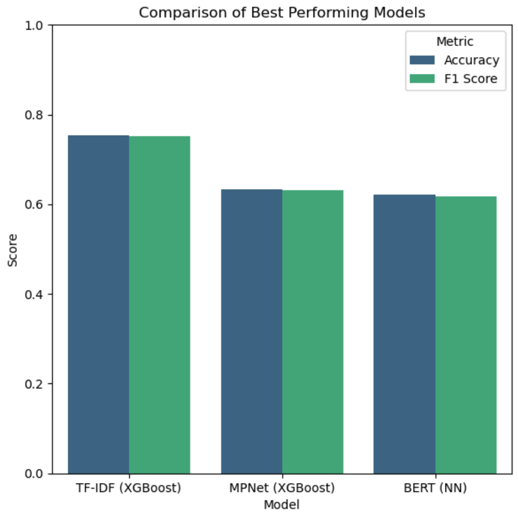

Figure 3: Best Performing Models by Vectorization/ Embedding Processes

The better performance by TF-IDF vectorization over advanced text embedding techniques like MPNet and BERT in job description classification could be due to several factors contributing to this observation. Firstly, the dataset's modest size and simplicity may limit the utility of sophisticated contextual understanding offered by models like BERT and MPNet. TF-IDF's reliance on basic term frequency calculations may provide a more reliable representation in such cases. Moreover, the task's straightforward nature and clear word associations in job descriptions might not demand the nuanced semantic grasp offered by advanced models. Therefore, TF-IDF's simplicity and interpretability may better suit the task's requirements, yielding best results. Additionally, the quality and diversity of the training data significantly impact advanced model performance. Noisy or insufficiently varied dataset, which might be the case because job descriptions usually follow a template, may hinder models' ability to learn meaningful representations, whereas TF-IDF may offer more robust features under such circumstances.

6. Conclusion

In conclusion, our project explored various methodologies for text classification in the domain of job description analysis, encompassing both traditional machine learning algorithms and advanced deep learning architectures, as well as different vectorization techniques including TF-IDF and word embedding transformers like MPNet and BERT. Among the methodologies assessed, XGBoost combined with TF-IDF vectorization emerged as the most effective approach, with classification accuracy and F1 score compared to more advanced word embedding techniques such as BERT and MPNet. Future work could focus on refining hyperparameters for advanced text embedding models like BERT and MPNet to further optimize their performance. Additionally, exploring domain-specific feature engineering by integrating industry-specific keywords and skill requisites into the classification process could enhance model accuracy and interpretability in job description classification.

References

Paul Trust, Provia Kadusabe, Haseeb Younis, Rosane Minghim, Evangelos Milios, and Ahmed Zahran. 2022. SNLP at TextGraphs 2022 Shared Task: Unsupervised Natural Language Premise Selection in Mathematical Texts Using Sentence-MPNet. In *Proceedings of*

TextGraphs-16: Graph-based Methods for Natural Language Processing, pages 119–123, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada. arXiv:2004.09297v2 [cs.CL] 2 Nov 2020.

Mayank Jobanputra and Lorena Martín Rodríguez. 2022. OversampledML at SemEval-2022 Task 8: When multilingual news similarity met Zero-shot approaches. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1171–1177, Seattle, United States. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2017. An Exploration of Word Embedding Initialization in Deep-Learning Tasks. In Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017), pages 56–64, Kolkata, India. NLP Association of India.

Mayukh Sharma, Ilanthenral Kandasamy, and W.b. Vasantha. 2021. YoungSheldon at SemEval-2021 Task 5: Fine-tuning Pre-trained Language Models for Toxic Spans Detection using Token classification Objective. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 953–959, Online. Association for Computational Linguistics.

Marco Di Giovanni, Thomas Tasca, and Marco Brambilla. 2022. DataScience-Polimi at SemEval-2022 Task 8: Stacking Language Models to Predict News Article Similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1229–1234, Seattle, United States. Association for Computational Linguistics.

Nidhi Goyal, Jushaan Kalra, Charu Sharma, Raghava Mutharaju, Niharika Sachdeva, and Ponnurangam Kumaraguru. 2023. JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2181–2191, Dubrovnik, Croatia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. 2024. Deep Learning-based Computational Job Market Analysis: A Survey on Skill Extraction and Classification from Job Postings. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 1–15, St. Julian's, Malta. Association for Computational Linguistics.

Hakan Ceylan and Yookyung Kim. 2009.

Language Identification of Search Engine Queries. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1066–1074, Suntec, Singapore. Association for Computational Linguistics.

Thomas Green, Diana Maynard, and Chenghua Lin. 2022. Development of a Benchmark Corpus to Support Entity Recognition in Job Descriptions. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1201–1208, Marseille, France. European Language Resources Association.

Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. 2020. Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5832–5842, Barcelona, Spain (Online). International Committee on Computational Linguistics.