COMP90049
Job Salary Prediction Report

Anonymous

1 Introduction

In the field of machine learning, the dominant paradigms, such as Naïve Bayes, KNN, and Logistic Regression, primarily revolve around supervised learning, which necessitates a substantial amount of labeled data. However, acquiring labeled data is often challenging due to its scarcity or high cost. Consequently, the incorporation of unlabelled data has garnered attention as a means to augment data volume and potentially improve model performance. Thus, this study seeks to explore the impact of unlabelled data on the predictive accuracy of job salary estimation, drawing a comparison between the advantages and limitations of supervised and semi-supervised learning approaches.

2 Literature Review

The dataset used in this research is generated from *Bhola et al*. (2020), and the blended data set are derived from Deri and Knight (2015), Das and Ghosh (2017), and Cook and Stevenson (2010). For machine learning purposes, two types of features were employed: TFIDF (term frequency - inverse document frequency) as described in *Schutze et al*. (2008), and a 384-dimensional embedding representation derived from each job description based on the work of Reimers and *Gurevych* (2019).

In recent decades, research into semi-supervised learning method in Natural Language Processing provides insights to utilize the unlabeled data in predictions. One strategy is multitasking learning, which can be a powerful tool for improving discourse classification in NLP, particularly in the context of small, class-imbalanced datasets (*Spangher et al*., 2021). Another method in utilizing the unlabeled data is to combine existing supervision with semi-supervised learning methods to improve both performance and interpretability of the prediction models (*Pryzant et al.,* 2022).

3 Method

3.1 Research Objective

The primary aim of this study is to conduct a comprehensive analysis of the potential improvement in machine learning tools' performance for salary predictions through the utilization of unlabeled data.

To accomplish this objective, the study incorporates self-training, a semi-supervised learning paradigm. The research process encompasses the entire spectrum of machine learning, beginning with the development of a model exhibiting optimal predictive performance using a labeled dataset. Subsequently, the self-training method is applied to the selected base estimator, thereby enabling the incorporation of all unlabeled data into the semi-supervised machine learning model. Ultimately, both supervised and semi-supervised models are assessed and scrutinized to draw conclusive insights regarding the impact of unlabeled data on salary prediction performance, based on the employed dataset.

3.2 Data Preprocessing

Given the research focus on unlabeled data and salary prediction performance, the training dataset is partitioned into two subsets based on the presence or absence of labels.

Closer examination of the labeled training data reveals an uneven distribution among the different salary bins, with the fifth bin exhibiting the lowest number of job descriptions. This discrepancy may potentially impact the performance of the model in accurately predicting instances belonging to bin 5 (Figure 1).

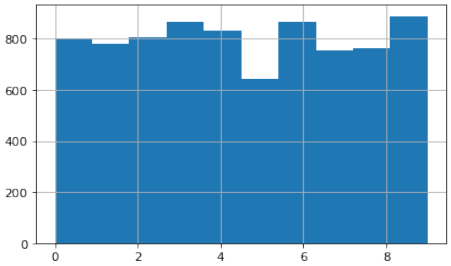
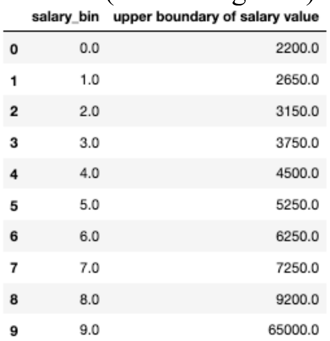


Figure 1- Number of instances in different salary bins in labelled training dataset.

Furthermore, it is noteworthy that the division of salary bins based on equal frequency results in a substantial variation in the range of mean salary values within each bin. This disparity becomes particularly evident when comparing the ranges across different bins (refer to Figure 2).



	salary_bin	upper boundary of salary value
0	0.0	2200.0
1	1.0	2650.0
2	2.0	3150.0
3	3.0	3750.0
4	4.0	4500.0
5	5.0	5250.0
6	6.0	6250.0
7	7.0	7250.0
8	8.0	9200.0
9	9.0	65000.0

Figure 2 - Upper boundary of salary value in each of the salary bins.

3.3 Model Selection

3.3.1 Baseline Models

In order to establish a benchmark for performance evaluation, two baseline models, namely Zero-R and Random Baseline, have been trained and assessed.

The inclusion of Zero-R is motivated by its capability to classify instances solely based on the most prevalent class observed in the training data, making it suitable for the structure of the labelled training data that was generated using an equal-frequency method. The adoption of the Random Baseline model is intended to facilitate a fair and impartial comparison with Zero-R, thereby establishing an unbiased benchmark for the subsequent model selection process.

The accuracy of these two baseline models is shown below (see Table 1).

	ZERO-R	**RANDOM**
TFIDF	0.099597006	0.101324122
EMBEDDING	0.099597006	0.099366724

Table 1- Accuracy of baseline models trained on TFIDF and Embedding features.

3.3.2 Candidate Models

Given the specific features and dataset utilized for job salary prediction, a comprehensive selection of candidate models has been compiled to address the problem at hand.

Several candidate machine learning models are employed, including Gaussian Naïve Bayes, Decision Tree, K-Nearest Neighbors (KNN), Perceptron, Logistic Regression, and Multilayer Perceptron (MLP). Gaussian Naïve Bayes is specifically chosen for its effectiveness in text classification tasks, particularly when using TF-IDF features (Raschka, S., 2014). The Decision Tree model is utilized to handle potential outliers and missing values in the dataset. KNN is a suitable choice due to its inherent capability to handle multi-class classification, which aligns with the objective of salary prediction in this dataset. Logistic Regression, Perceptron, and MLP are selected based on their ability to analyze relationships among correlated features, thereby potentially enhancing the predictive performance of the models.

3.3.3 Parameter Tuning

In the process of parameter tuning for optimizing the performance of selected models in terms of accuracy and time complexity, the method of control variates is employed. This approach involves training the model iteratively, adjusting the parameters in each iteration to find the configuration that yields the best performance and generalization.

3.4 Evaluation Metrics

The evaluation metrics employed for model selection encompass both accuracy and time complexity. Accuracy measures the predictive correctness of the models, providing an assessment of their overall performance. Time complexity, on the other hand, offers insights into the computational efficiency and resource requirements of the models, allowing for considerations of their practical feasibility.

In addition, the use of confusion matrices is also employed to provide a detailed performance analysis of the models. Confusion matrices offer a comprehensive breakdown of the model's predictions, allowing for a deeper understanding of its performance across different classes and providing insights into any potential biases or misclassifications.

4 Results

4.1 Model Selection

The accuracy and time complexity results of the candidate models are presented in Figure 3 and Figure 4, respectively.

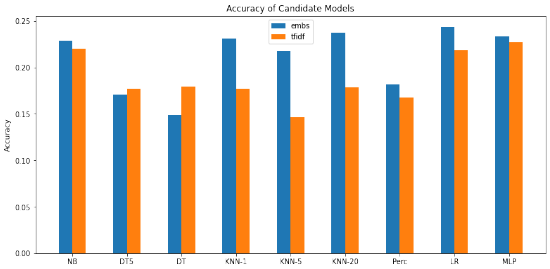


Figure 3- Accuracy of candidate models trained on embedding and TFIDF features.

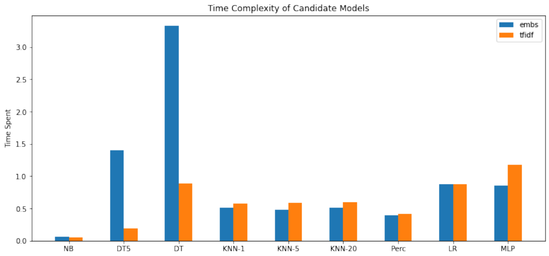


Figure 4- Time Complexity of candidate models trained on embedding and TFIDF features.

Among the candidate models, Logistic Regression, KNN, MLP, and Naïve Bayes exhibit promising performance in terms of accuracy. Notably, models trained on the embedding feature demonstrate higher accuracy compared to those trained on TFIDF. Additionally, KNN and logistic regression models demonstrate lower time complexity compared to the other models within this subset.

Consequently, KNN and logistic regression are selected as the supervised learning models for this research. Further, parameter tuning is applied to optimize these selected models, ensuring the best fit for the given task.

4.2 Parameter Tuning

In this phase, parameters in both KNN and logistic regression models are tuned to achieve their best performance in accuracy and time complexity.

4.2.1 Selected model 1 – KNN

The K-Nearest Neighbor (KNN) model is a classification algorithm that classifies data based on the majority class of k-nearest training examples in feature space. The performance of the KNN model is highly dependent on the number of neighbors and the distance metric. In this research, these two parameters are tuned to optimize the KNN model and balance its complexity. The method of control variates is employed as a systematic experimentation with different parameter settings to identify the optimal combination in this step.

Figures 5 and 6 demonstrate the accuracy and time complexity of KNN models with different numbers of neighbors ranging from 1 to 30. The accuracy of the KNN model reaches its peak when k is equal to 20 in embedding feature, which also provides the best time complexity performance on the validation set.

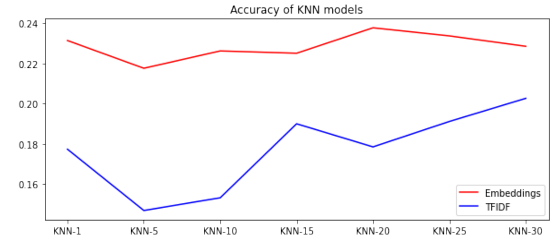


Figure 5 - Accuracy of KNN models with different number of neighbors (k).

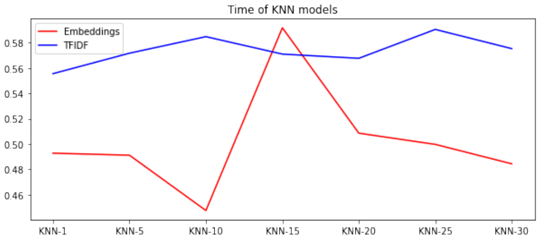


Figure 6 – Time Complexity of KNN models with different number of neighbors (k).

Figures 7 and 8 show the performance of KNN models on different distance metrics. The KNN model trained on embedding features obtains the best performance with Manhattan distance, while the model with TFIDF performs better on Jaccard distance.

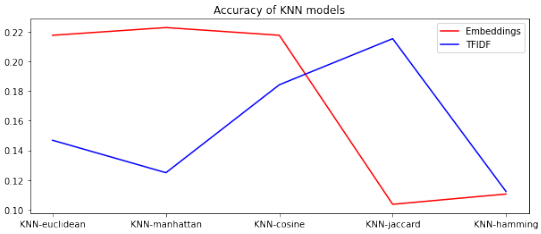


Figure 7 - Accuracy of KNN models with different distance metrics.

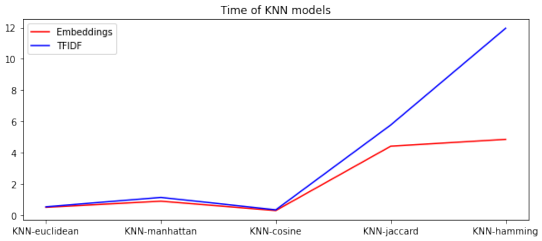


Figure 8 - Time of KNN models with different distance metrics.

Consequently, the k value of 20 and Manhattan distance are applied as the parameters in my KNN model.

4.2.2 Selected model 2 – Logistic Regression

When tuning the logistic regression model for predicting job salaries, the key parameters to consider are the solver, which estimates the coefficients that best fit the data, and the regularization techniques, which prevent the model from overfitting and improve regularization.

In Figure 9, I observe that all solvers provide reasonable accuracy in embedding features. However, the time spent varies with different solvers, with 'lbfgs' demonstrating both high accuracy and lowest time complexity (as seen in Figure 10). In addition, the 'lbfgs' solver is suitable for multiclass classification problems and performs well with larger datasets in salary prediction. Hence, the 'lbfgs' solver is used as the solver of the prediction model.

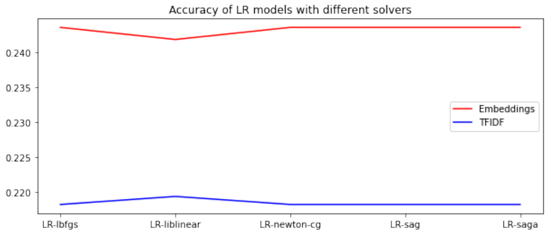


Figure 9 - Accuracy of LR models with different solvers.

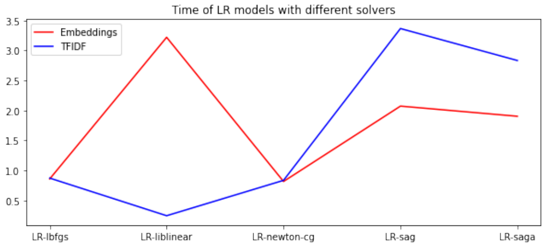


Figure 10 - Time of LR models with different solvers.

To optimize the generalization of the logistic regression model, the parameter C (regularization strength) is tuned to control the balance between model simplicity and fitting to the training data. The experiment revealed that the model achieved its best accuracy performance when C was equal to 0.55. Therefore, this value is utilized in the logistic regression model. By tuning the solver and regularization parameter, the model's generalization and predictive performance are optimized, ensuring that it is well-suited for the task of predicting job salaries.

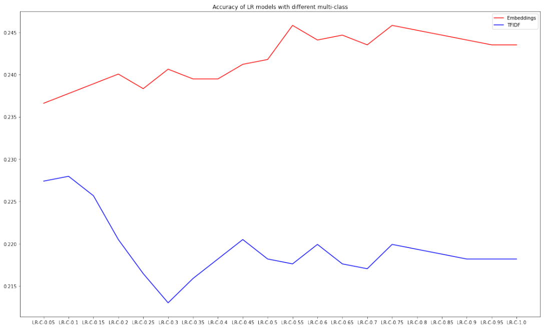


Figure 11 - Accuracy of LR models with different penalties.

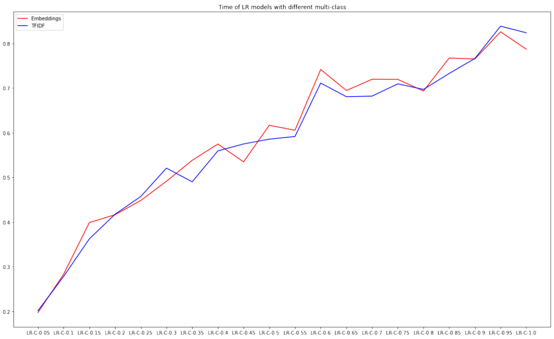


Figure 12 - Time of LR models with different penalties.

4.3 Semi-supervised Learning

The self-training technique, which is a form of semi-supervised learning, is employed in this study. The K-Nearest Neighbor (KNN) model and logistic regression model on embedding feature are selected as the base estimators for self-training.

The effectiveness of the self-training technique is largely contingent on the quality of the generated labels. To assess whether the existing unlabeled data can enhance salary prediction, self-training models are trained with different thresholds and compared to the base models. The results indicate that the self-training technique does not lead to performance improvement in the logistic regression (LR) model (see Figure 13). However, it demonstrates superior performance when applied to the KNN model (see Figure 14).

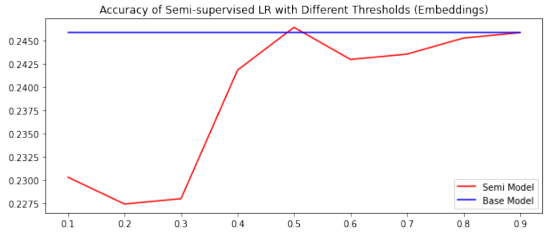


Figure 13 - Accuracy of self-training LR model with different thresholds.

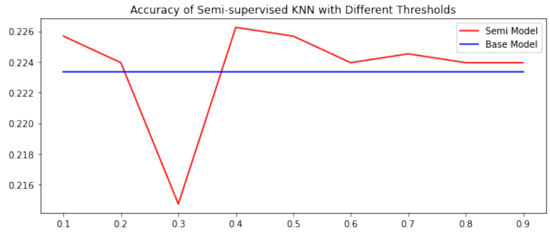


Figure 14 - Accuracy of self-training KNN model with different thresholds.

5 Discussions & Critical Analysis

5.1 Semi-supervised Learning

The findings from the experiments on semi-supervised learning, presented in Figures 13 and 14, suggest that the effectiveness of the self-training technique may depend on the choice of the base estimator. Specifically, the KNN model exhibits a significant improvement in performance when leveraging the additional information provided by the unlabeled data, whereas the LR model does not exhibit the same level of improvement.

It is worth noting that the self-training technique may be more beneficial for poorer models compared to better-performing models, as demonstrated in this study where the LR model initially outperformed the KNN model in terms of accuracy and generalization. This may be attributed to the fact that self-training can exploit the additional information contained in the unlabeled data, which can enhance the model's ability to capture the underlying data distribution.

One possible explanation for the observed improvement in the KNN model performance is that the model's assumption of similarity among instances based on their distance can benefit from leveraging a larger pool of unlabeled data. The labeled data alone may not be sufficient for the KNN model to generate accurate predictions, but incorporating the unlabeled data can help overcome this limitation.

Moreover, the iterative process of self-training can help correct errors as the model generates labels for the unlabeled data based on its current predictions. While these labels may contain some errors, the model can learn from its mistakes and update its parameters accordingly. By iteratively training and refining the model on the combined labeled and pseudo-labeled data, the KNN model has the opportunity to correct its errors and improve its predictions over time.

In this research, even though the unlabeled data helps improve the performance of KNN model by increasing the accuracy to approximately 22.625% with threshold as 0.4, its performance is still weaker than logistic regression model trained on labelled data with embedding feature, whose accuracy is 24.583%. Moreover, the time complexity of logistic regression model is lower than KNN. Therefore, the final selected model is logistic regression with ‘lbfgs’ solver, ‘l2’ penalty, and 0.55 as the Inverse of regularization strength.

5.2 Evaluation Metric

In addition to standard metrics such as accuracy and time complexity, the performance of the final logistic regression model is also evaluated using a confusion matrix, which provides a more detailed summary of its predictive capabilities. As shown in Figure 15, the model is able to accurately predict salaries within a reasonable range, with high values along the diagonal indicating a strong agreement between predicted and true labels.

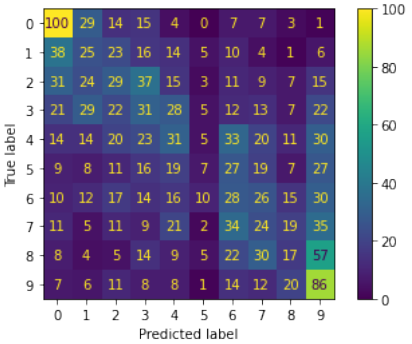


Figure 15 - Confusion matrix of LR model

However, due to the uneven distribution of instances across salary bins in the labeled training data (as observed in Figure 1), the model's accuracy is lower when predicting instances in bin 5. Moreover, the range of mean salary values in each category varies greatly, particularly in bins 0 and 9, which results in extremely high accuracy in these bins.

Overall, the confusion matrix provides valuable insights into the performance of the final salary prediction tool and indicates that it meets the required level of accuracy for its intended use.

6 Conclusions

Overall, the study has investigated and analyzed the performance of various machine learning models for salary prediction. Self-training method has been experimented to provide an insight on the improvement of job salary prediction with unlabeled dataset: the semi-supervised learning technique can help improve the KNN model that requires more information from training data but provides little improvements for LR model in this research. However, further studies are required to figure out the underlying issue when employing self-training methods and then generate the solution to improve the salary prediction.

7 References

Cook, P. and Stevenson, S. (2010) Automatically identifying the source words of lexical blends in English. *Computational Linguistics*, 36(1): 129–149.

Das, K. and Neuramanteau, S. (2017) A neural network ensemble model for lexical blends. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 576–583. Taipei, Taiwan.

Deri, A. and Knight, K. (2015) How to make a frenemy: Multitape FSTs for portmanteau generation. In *Human Language Technologies: The2015 Annual Conference of the North American Conference of the North American Chapter of the ACL, pages 206–210. Denver, USA.*

Eisenstein, J., O'Connor, B., Smith, N. A. and Xing, E. P. (2010) A latent variable model for geographic lexical variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pages 1277–1287. Cambridge, USA.

Pryzant, R., Yang, Z., Xu, Y., Zhu, C., & Zeng, M. (2022). Automatic rule induction for efficient semi-supervised learning. arXiv preprint arXiv:2205.09067.

Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. arXiv preprint arXiv:1410.5329.

Spangher, A., May, J., Shiang, S.-R., & Deng, L. (2021). Multitask Semi-Supervised Learning for Class-Imbalanced Discourse Classification. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/2021.emnlp-main.40