Exploring the impact of machine learning algorithms with unlabelled data

Anonymous

1 Introduction

Predicting salary based on job descriptions is a challenging task in the field of natural language processing and machine learning. In the current digital age, many recruiters seek to find suitable candidates through multiple channels — e.g., online job portals, professional networks — as well as traditional avenues, such as word of mouth and mass media (Shenoy and Aithal, 2018).

The dataset is derived from the large dataset called *mycareersfuture* (Bhola et al., 2020). The dataset has a total of 17377 data, consisting of 13902 train data, 1738 validation data, and 1737 test data. The dataset is shown in the table 1:

Table 1: Dataset Information

Data Type	Labeled	Unlabeled	Total
Train	8000	5902	13902
Validation	1738	-	1738
Test	-	1737	1737
Total	9738	7639	17377

The distribution of salary bin is shown in the figure 1. We observe that the salary bin distribution exhibits an uneven and imbalanced pattern, which may potentially affect the performance of the machine learning algorithms.

To answer the question ”Does Unlabelled data improve Job salary prediction?”, We will analyse and compare the performance of different machine learning algorithms for this dataset (labelled and unlabelled data) and finally explore whether unlabelled data can be effectively combined to increase the performance of the model.

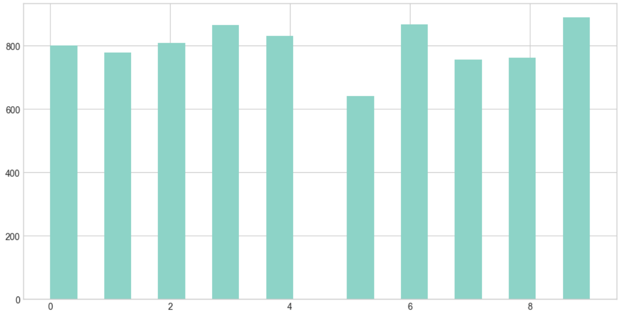

Figure 1. Salary Bin Distribution

2 Literature review

3 Methods

In this study, we adopt two feature representations from the raw job descriptions.

- **TF-IDF**: We compute the TF-IDF vectors for job descriptions using the method proposed by (Manning et al., 2008). This method captures the importance of terms within a document and across the entire corpus.
- **Embedding**: We adopt the pretrained Sentence Transformer model (Reimers and Gurevych, 2019) to obtain word embeddings for the job descriptions. These embeddings provide semantic representation for the text data.

Through these two features, we explore three different machine learning algorithms paradigms: Supervised learning, Unsupervised learning and Semi-supervised learning.

To find the parameters which can lead the highest performance of the machine learning algorithms, we consider to use some search strategies. Due to a high dimentions of the features, we adopt Grid search here because Grid search can suffer from high dimensional spaces (Liashchynskyi and Liashchynskyi, 2019).

We train the models between TFIDF and Embedding features, but we will only choose TFIDF features to analyse the results and evaluate the models. To evaluate classifiers, we use the F_1 score that combines recall and precision in the following way: (Tan, 2006)

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

3.1 Supervised learning

In the supervised learning part, we adopt 9 different machine learning algorithms to predict the salary bin.

3.1.1 KNN Classifier

We decide to use k-Nearest-Neighbours (kNN) as our baseline model, because the k-Nearest-Neighbours is a simple but effective method for classification (Guo et al., 2003). To ensure what weights and p value in KNN classifier lead to a better performance, We set k value in the range of 1 to 11, and weight options are uniform and distance, p value is 1 and 2 (represent manhattan distance and euclidean distance).

Using Grid search, we find that in TF-IDF, using kNN algorithm's best accuracy is 18.77% with parameters (k = 3, p = 2, weights = "distance"). In Embedding, using kNN algorithm's best accuracy is 23.95 % with parameters (k = 3, p = 2, weights = "distance").

In the experiment, we found that K is the most important factor in kNN algorithm. So we keep increasing k's value to 200. And set p equal to 2 and weight is "distance". The output is shown in figure 2.

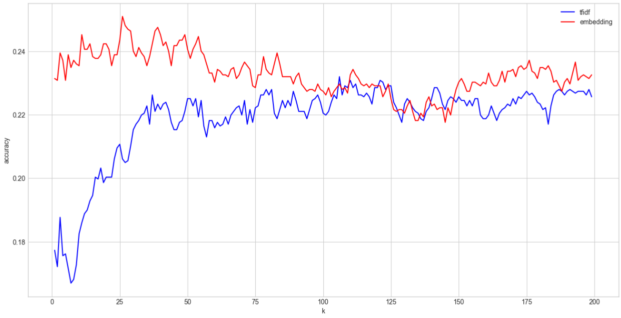

Figure 2. KNN Classifier

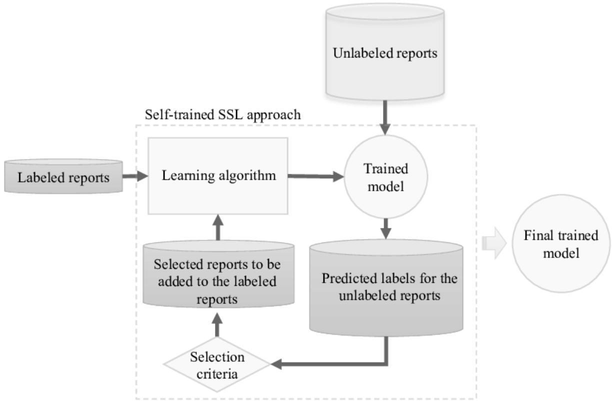
Unlabeled reports
Self-trained SSL approach
Labeled reports
Learning algorithm
Trained model
Final trained model
Selected reports to be added to the labeled reports
Predicted labels for the unlabeled reports
Selection criteria

Figure 3. Self-trained semi-supervised learning architecture (Hassanzadeh et al., 2018)

From the results, we can when the k value is 106, the accuracy for tfidf is 23.20% which is the best accuracy for tfidf. For embedding data, the best accuracy is 25.1% when k is 26.

3.1.2 Decision Tree Classifier

Decision It focuses on generating classification rules displayed as decision trees that is deduced or concluded from a group of disorder and irregular instances. (Dutta et al., 2018)

3.1.3 Naive Bayes Classifier

3.1.4 Other Classifier

3.2 Unsupervised learning

3.3 Semi-supervised learning

we will use the architecture shown in figure 3 to train the model.

4 Results

5 Discussion / Critical Analysis

6 Conclusions

References

Bhola, A., Halder, K., Prasad, A., and Kan, M.-Y. (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.

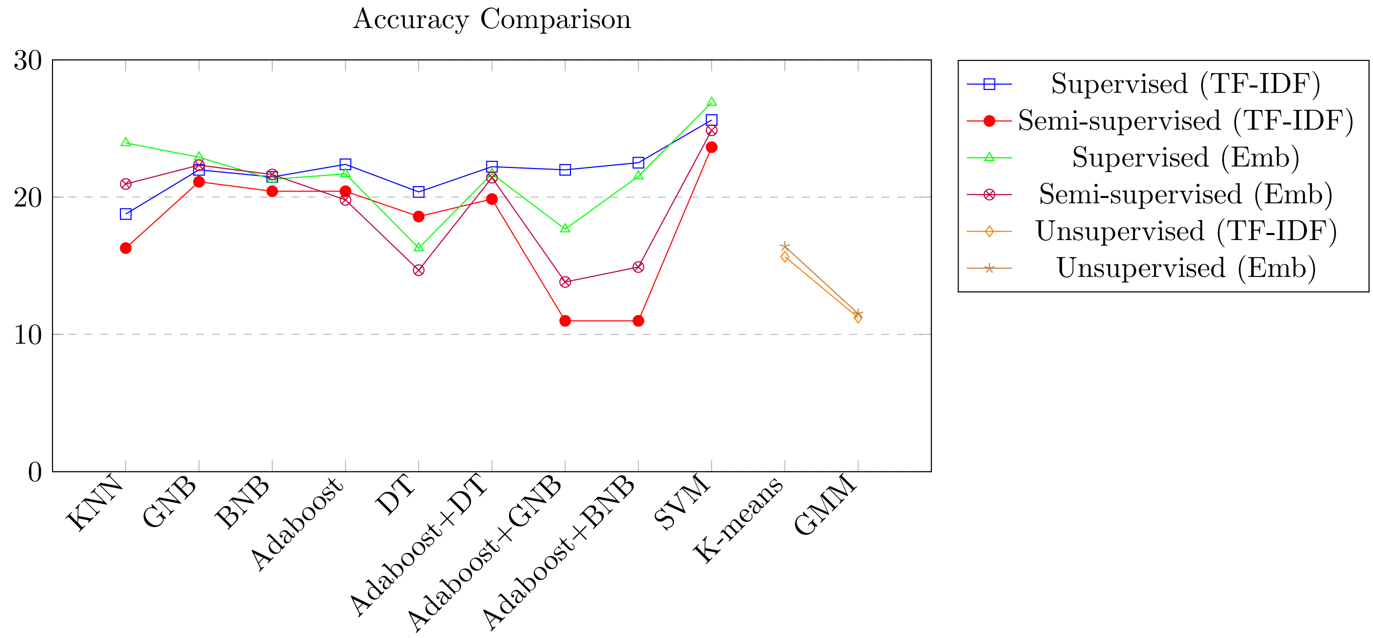

Table 2: Accuracy Comparison

Model	TFIDF		Embedding	
	Labaled data	Unlabeled data	Labaled data	Unlabaled data
KNN	18.77	16.29	23.95	20.96
GNB	21.99	21.12	22.91	22.33
BNB	21.47	20.43	21.3	21.65
Adaboost	22.39	20.43	21.7	19.8
Decision Tree (DT)	20.38	18.59	16.29	14.68
Adaboost + DT	22.22	19.86	21.7	21.42
Adaboost + GNB	21.99	10.99	17.67	13.82
Adaboost + BNB	22.51	10.99	21.53	14.91
SVM	**25.62***	**24.64***	**26.89***	**24.87***
K-means	-	**15.66***	-	**16.41***
GMM	-	11.23	-	11.51

Dutta, S., Halder, A., and Dasgupta, K. (2018). Design of a novel prediction engine for predicting suitable salary for a job. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 275–279.

Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. (2003). Knn model-based approach in classification. In Meersman, R., Tari, Z., and Schmidt, D. C., editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 986–996, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hassanzadeh, H., Kholghi, M., Nguyen, A., and Chu, K. (2018). Clinical document classification using labeled and unlabeled data across hospitals.

Liashchynskyi, P. and Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for nas.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. An Introduction to Information Retrieval. Cambridge University Press.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Shenoy, V. and Aithal, S. (2018). Literature review on primary organizational recruitment sources. *International Journal of Management, Technology, and Social Sciences*, pages 37–58.

Tan, S. (2006). An effective refinement strategy for knn text classifier. *Expert Systems with Applications*, 30(2):290–298.

Zhang, J. and Cheng, J. (2019/08). Study of employment salary forecast using knn algorithm. In *Proceedings of the 2019 International Conference on Modeling, Simulation and Big Data Analysis (MSBDA 2019)*, pages 166–170. Atlantis Press.