Person-job-fit prediction with deep neural network models based on various pre-trained models.

Anonymous Author(s)**

ABSTRACT

Finding a job is suitable for a person looking for a job, based on their job descriptions,: In this paper, we focus on examining person profile using various deep neural network models including TextCNN, CNN-LST, TextRNN, and RCNN with various pre-trained word embeddings in the job data set. We also proposed a simple and effective ensemble model that combines different deep neural network models. Our experimental results showed that our proposed ensemble model had the highest score with an F1 score of 89%. Furthermore, we analyze these experimental results to understand this problem in order to find better solutions in the future.

In this work we implement and benchmark various deep neural network language models to find suitable jobs for a person with a given skillset. The neural networks include TextCNN, CNN-LST, TextRNN, and RCNN with various pre-trained word embeddings. We refine these models using a dataset we collected from German online job boards. In addition we propose an effective ensemble model combining the best performing deep learning architectures for this specific task. Our proposed model performs with a F1-Score of 89% on the 89-class classification task. Furthermore, we analyze these experimental results to understand this problem in order to find better solutions in the future.

KEYWORDS

neural networks, text classification, ensemble methods, human resources

ACM Reference Format:

Anonymous Author(s). 2022. Person-job-fit prediction with deep neural network models based on various pre-trained models.. In *Madrid 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid 2022.* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The strong development of the labor market has led to a large number of job offers and the requirements of the respective occupation in recent years. Because of the diversity, students or job seekers want a method of finding a job that corresponds to the knowledge and skills they have acquired in school or in the work process. Also, the recruiting company must manually filter the candidate profiles to select the people who are suitable for the position being recruited, which is time consuming, while the number of applications can reach hundreds or thousands. Hence, we would like to investigate the task of job prediction to help you effectively address the above issues.

ACM SIGIR 2022, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 RELATED WORK

TextCNN demonstrates its effectiveness in machine learning in general and in natural language processing in particular. Firstly, we want to mention the CNN model for a text classification problem [] This model has been used in experiments and evaluated on a variety of datasets such as MR [], SST-1, SST- [], Subj [], TREC [], CR [], MPQA [], VLSP-2018 [] and UIT-VSMEC [8] where it yields good performances. The model architecture can also be used in Hate Speech Detection [], [] yielding good results. This TextCNN also proves effective with problems similar to image classification []. In order to increase the model's predictive results, words are represented in a vector space to express well the semantic relationship between words. Commonly used using pre-trained word embeddings [] are Glove [] and FastText [].

Training a machine learning model with a classification task that has a large number of prediction classes is challenging. One possible approach is the classifier ensemble methods combining multiple outputs of multiple models to increase the performance of the prediction. There are many basic [], [] to advanced ensemble methods [], []. These methods vary in the way the multiple model outputs are combined. However, in this paper, we use the majority voting method. This method is quite simple and recently proved effective in text classification problems []–[].

3 DATASET

The dataset used in the study consists of 350,000 different job descriptions collected from online job search sites. We annotated the data with 89 different types of related jobs. These categories mainly relate to roles that are typically found in today's data-driven economy. The table shows the distribution of job labels in our data set. The job posting rate for each day is 1.20% to 5.11%. The data set is relatively balanced. However, there is a lot of overlapping information between the descriptions of these jobs for similar job titles making the classes intrinsically correlated. This poses a challenge for the model to differentiate between very similar job titles that have similar job descriptions and required skills. This ambiguity can be overcome when combining very similar job titles into overarching job classes.

4 METHODOLOGY

In this study we implemented four deep neural networks Models for occupational classification and proposal of an ensemble model. In addition, we implement two pre-trained word embeddings in these models.

TextCNN:. motivated from [], text for classifying different occupational classes. TextCNN shows great promise to deliver the desired result. in our case, we modified TextCNN architecture to suit multiple classes (89 classes) classifications.

Text CNN consists of three main parts (convolution layer, pooling layer, and fully connected layer). The convolution layer consists of filter types with a total of 1024 filters for extracting high-level features. These then pass through the pooling layer, which is responsible for reducing the spatial size of the folded feature and reducing the computing power required to process the data by reducing the dimensions. The convolutional layer and the pooling layer together form the i-th layer of a convolutional neural network. As the process continues, the final output is flattened and fed into a normal neural network in the fully connected layer for classification purposes using the Softmax classification technique.

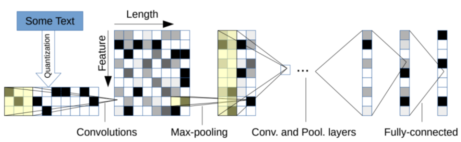

Figure 1: TEXTCNN

TextRNN/TextBiRNN :. This section reviews some of the most prominent attention models which create new state of the arts on TC tasks, when they are published. Yang et al. [79] propose a hierarchical attention network for text classification. This model has two distinctive characteristics: (1) a hierarchical structure that mirrors the hierarchical structure of documents, and (2) two levels of attention mechanisms applied at the word and sentence-level, enabling it to attend differentially to more and less important content when constructing the document representation. This model outperforms previous methods by a substantial margin on six TC tasks. Zhou et al. [80] extend the hierarchical attention model to cross-lingual sentiment classification. In each language, a LSTM network is used to model the documents. Then, classification is achieved by using a hierarchical attention mechanism, where the sentence-level attention model learns which sentences of a document are more important for determining the overall sentiment. while the word-level attention model learns which words in each sentence are decisive. Shen et al. [81] present a directional self-attention network for RNN/CNN-free language understanding, where the attention between elements from input sequence(s) is directional and multi-dimensional. A light-weight neural net is used to learn sentence embedding, solely based on the proposed attention without any RNN/CNN structure. Liu et al. [82] present a LSTM model with inner-attention for NLI. This model uses a two-stage process to encode a sentence. Firstly, average pooling is used over word-level Bi-LSTMs to generate a first stage sentence representation. Secondly, attention mechanism is employed to replace average pooling on the same sentence for better representations. The sentence's first-stage representation is used to attend words appeared in itself.

TextCNN-LSMT:. Recurrent Neural Network(RNN) has been widely used in the processing of variable-length sequences. However , since the typical RNN is equivalent to multi-layer feed-forward neural networks, a large amount of historical information brought by long sequences will bring about vanishing gradient and information loss. Long Short-Term Memory(LSTM) is an improved network structure designed based on RNN. LSTM effectively saves the history information in long sequences by adding memory cell and three control gates, which improves the loss of historical information and gradient vanish caused by excessive layer RNN training.

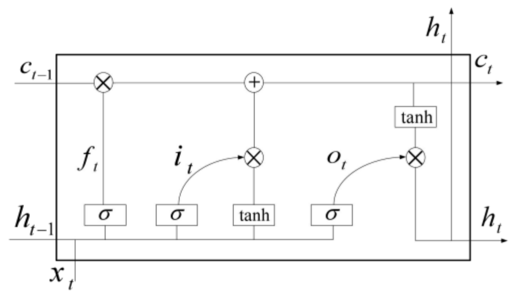

Figure 2: LSTM Cell Structure

The LSTM structure is shown is Figure []. The structure adds a memory cell for storing history information, and the update, deletion and output of history information are controlled by three gates respectively,they are input gate, forget gate and output gate. The input gate is used to determine how incoming vectors alter the state of the memory cell. The output gate allows the memory cell to have an effect on the outputs. In the end, the forget gate can allow the memory cell to remember or forget its previous information.

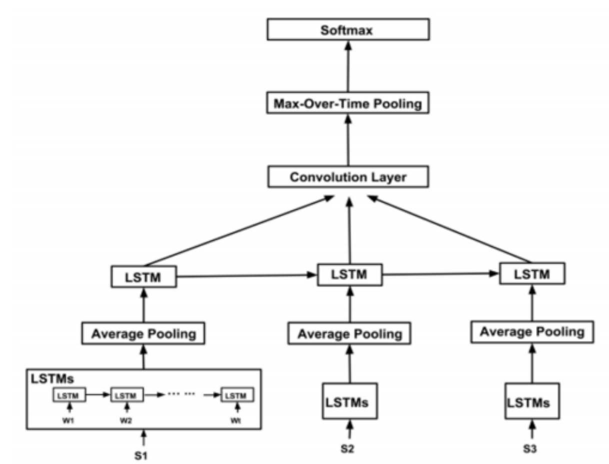

Figure 3: CNN-LSTM

RCNN:. We propose a deep neural model to capture the semantics of the text. Figure [] shows the network structure of our model. The input of the network is a document D, which is a sequence of words $w_1, w_2 ... w_n$. The output of the network contains class elements.

We use to denote the probability of the document being class k, where Θ is the parameters in the network. rectional recurrent neural network, to capture the contexts.

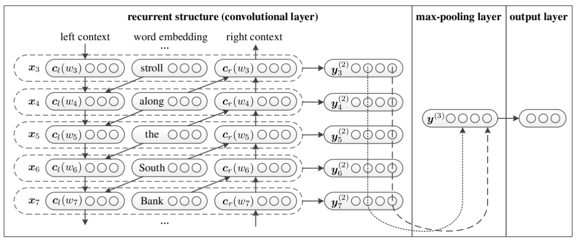

Figure 4: RCNN

We define $c_l(w_i)$ as the left context of word w_i and $c_r(w_i)$ as the right context of word w_i . Both $c_l(w_i)$ and $c_r(w_i)$ are dense vectors with $|c|$ real value elements.

5 EXPERIMENTS

In this study, we implement several simple and effective

- techniques to pre-process data for the model's input as follows.
- Converting the job descriptions into the lowercase strings.
- Deleting special characters such as # & , etc.
- Segmenting job descriptions into a set of words
- Removing the stop word in the descriptions
- Representing words into vectors with pre-trained word embedding sets.

In this study, we split randomly the dataset into three different sets including 10% for the testing set, 90% for training set. In the training set, we get 20% for the validation set. To evaluate our models, we use four measures of measurement such as accuracy. We conducted a series of experiments and our experimental results are shown in Table. the result of this model achieves an accuracy of 89.40%, along with this model also gives impressive results in other measurements such as 88.38%. The LSTM architecture combine with CNN has been more effective than the TextCNN architecture when combined with the CNN model. Although not the best model in the test set, the AttCNN model combined with the pre-trained word embeddings such as Glove and FastText also produced quite high results. Finally, we conduct experiments on our proposed ensemble model. We found that the ensemble method achieve the best results and was much higher than other models. It can be seen that the ensemble method has stable results in all metrics

6 CONCLUSION AND FUTURE WORK

In this paper, we implemented the TextCNN model and more complex models such as and

with various word embeddings to solve the job prediction. From experimental results of the deep neural networks and leveraging the power of each model,

We have conducted a number of experiments The results are shown in the third table. What particularly surprised us was that the TEXT CNN with Embedding Glove proved to perform well and

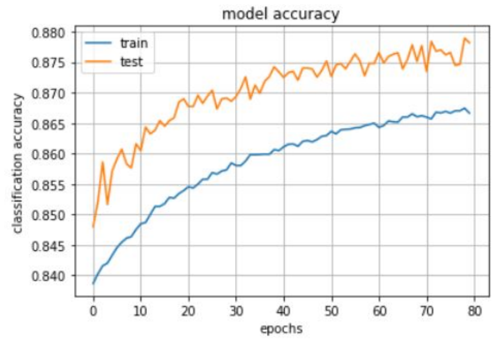

Figure 5: te

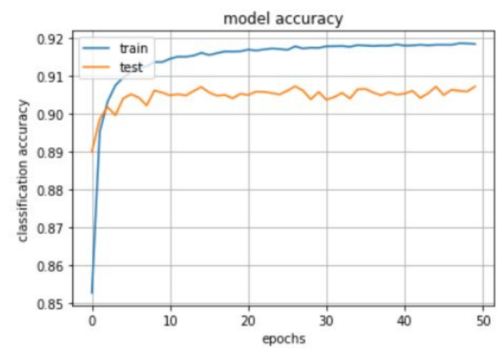

Figure 6: te

outperformed the other models (LSTM-CNN and RCNN models). This model achieves an accuracy of 89%, and this model also delivers impressive results for other measurements.

B. 72.38% for F1 classification, 72.46 % for accuracy and 72.30% call again.RCNN architecture more effective than the CNN-LSTM Along with the TEXT CNN model. The RCNN model wasn't the best model in the test set, Experiments with the highest accuracy rating of 66.00%. TextCNN model [28] compared to TEXT-RNN binary model LSTMCNN with word embedding FastText for better accuracy 71.20% and model LSTM-CNN with the word Embedding glove with 70.30% accuracy. Finally, we conduct experiments on our proposed group. Model. We found that the clustering method gave the best results and was much higher than the other models at 72.70for precision, 72.71% for F1, 72.83% for precision and 72.59% for reimbursement.

———

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

7 TABLES

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's

7.1 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\rightarrow \infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_{0}^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

8 FIGURES

Table 2: Some Typical Commands

Command	A Number	Comments
`\author`	100	Author
`\table`	300	For tables
`\table*`	400	For wider tables