Salary Prediction using Bidirectional-GRU-CNN Model

Zhongsheng Wang Shinsuke Sugaya Dat P.T. Nguyen
BizReach AI Technology Group
{zhongsheng.wang, shinsuke.sugaya, dat.nguyen}@bizreach.co.jp

1 Introduction

Information asymmetry between job seekers and employers is a long-standing problem. The prospective candidates generally have less knowledge about the actual treatment during interview and only at the final stages of the interview process they are actually informed of concrete offers. Meanwhile, it is vital for the employers to correctly guess the expectations of the candidates for crafting HR strategy; too low offers could lead to high decline rate and longer vacancies in the positions whilst, offering too much could result in high personnel expenses. Thus, it will be beneficial for most of us (except for sweatshop) to know the unbiased "market price" of the job positions, so that we can reduce mismatches and unsuccessful interviews. In this paper, we try to address this challenging problem by utilizing large-scale data combined with the state-of-the-art deep learning technologies. For this, we employ a dataset from Stanby[1], which is a search engine specializing in job information provided by BizReach Inc.[2], to train our salary prediction model. Among various deep structured language models, we implement bidirectional-GRU-CNN model, to predict the expected. It is observed that our proposed model outperforms the existing other models besides TextCNN, RCNN, BidLSTM and ResNet. Hence, it is a novel application on salary prediction text regression task with a combination deep learning neural network. Thus, our contribution in this paper is two-fold:

- We train an efficient deep learning model to predict the salary from the information posted on the web, and deploy it into real service.
- Given the success of convolutional neural architecture in language processing tasks, we propose a combination of deep architectures, Bidirectional-GRU-CNN to this end, which allows us to achieve higher accuracy than other competitive models.

The organization of this paper is as follows: In Section 2, we briefly reviews related works regarding salary prediction and recent breakthroughs in deep neural language models. In Section 3, we propose and explain our architecture, Bidirectional-GRU-CNN for regression task. Section 4 reports our experimental settings followed with the results in Section 5. Section 6 concludes this series of research with a future direction.

2 Related works

2.1 Salary prediction tasks

It is noteworthy that Adzuna once has held a competition on Kaggle[3] predicting the salary from the job contents. But salary has not been a very active area of research, presumably due to rather limited amount of available training data. To the best of our knowledge, very limited work has been carried out so far. For example, Nummi et al. proposed traditional modeling to predict individual income based on Finland dataset [13]. Li et al. and Jackman et al. proposed multi models for job salary prediction with text data [8,11]. However, the main concerns in these works are to test different types of algorithms, and then ensemble different results, rather than to train single effective model which can be of practical use in real service.

2.2 Text CNN

Neural network has been widely used to improve model performance of a various natural language processing (NLP) tasks. While RNN-like architectures have been dominant in most of NLP tasks; Yoon Kim proposed to use convolutional neural network (CNN), which was developed in the context of computer vision, in the sentence classification [9]. Even though CNNs may lose some important context information, they still are powerful on feature representation. Given the above success, CNN has been utilized not only in the text classification tasks but also in the machine translation [4] tasks.

2.3 RNN-CNN

Though, CNN's efficiency and sophisticated feature extraction mechanism are appealing, the inability to incorporate context information is obvious. Since the context information is very important on NLP problems, if it is possible to add context feature into CNN models, a better performance can be expected. Lai et al. proposed a RNN-CNN for text classification model [10], which first use RNN to learn context features and then feed those context-aware vectors into CNN.

[1] https://jp.stanby.com

[2] https://www.bizreach.co.jp

[3] https://www.kaggle.com/c/job-salary-prediction

Besides, we also take some other combination models as reference. For example, Schuster proposed bidirectional recurrent neural on classification experiments [15]. Rui Lu et al. proposed a Bidirectional-GRU model on sound detection [14]. Zhou et al. proposed a Bidirectional-LSTM on text classification [16]. Chiu et al. and Huang et al. proposed BI-LSTM-CNN [2] and BI-LSTM-CRF models [7] based on BI-RNN structure with further improvement on Named Entity Recognition (NER) and sequence tagging tasks.

3 Proposed Model

In this section, we propose a deep neural model for the prediction of annual salary by job description data posted on web. Figure 1 shows the front part of our network. As an input, the network receives a job description post, which includes title, contents, requirements, working time, job location, and job type and salary. The output of the front network contains *context* elements. Figure 2 shows the network structure back part of our model. We use a combination deep learning regression model to predict test job posting salary. In the following subsections, we explain our proposed model in detail.

3.1 Bidirectional-GRU Layer

3.1.1 Gated Recurrent Unit

Recurrent Neural Network is widely used in NLP field, which can learn context information of one word. Long Short Term Memory is designed to solve RNN gradient vanishing problem, especially learning long sentence [6]. Grate Recurrent Unit is a simplified LSTM cell structure [3]. Taking advantage of its simple cell, GRU can get a close performance to LSTM with less time.

3.1.2 Bidirectional-RNN Network

As stated in the previous section, even though TextCNN exhibits strength in feature representation, context feature cannot be incorporated very well by multi-size kernels. To combine a word and its context together, we first process words using a bidirectional-RNN network in the following way. One word in a sentence can be learn from forward and backward twice in bidirectional structure, which can get more word representation features and avoid some gradient vanishing problem in long sentence. For comparing performance of RNN, LSTM and GRU, we design in a series of models in our work.

3.2 Multi-channel Convolutional Layer

TextCNN apply multi-channel and different kernel size one dimensional convolutional structure on text data classification [9]. In this work, considering our data particularity, we apply a series of kernel size 1, 2, 3, 5, which is different from 3, 4, 5 kernel size in TextCNN. Different from semantic representation on classification task, some keyword representation is more important on job information regression.

After bidirectional-RNN learning context information, multi-channel convolutional layer can learn better features than TextCNN.

3.3 Fully Connection Layer

After concatenate multi-channel CNN output, we use a fully connection layer to finish regression work. Between each of the dense layer, we add dropout and batch normalization layer to avoid gradient vanishing.

3.4 Pre-training word vectors

Pre-training word level vector already is a basic part in deep learning model since Word2Vec [12]. In this work, we choose FastText [1] as our pre-training model. We use all the 8 million job posting data to train our word vector because of its specific domain.

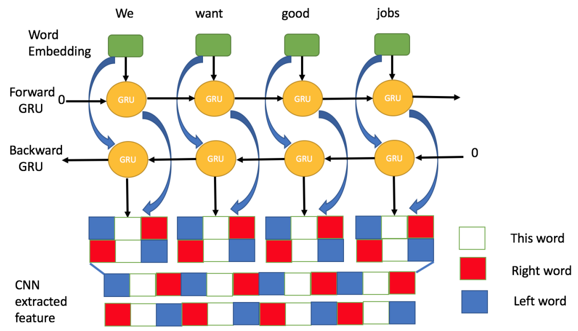

Figure 1: Bidirectional-GRU-CNN model

Bid-GRU-CNN model structure shows in

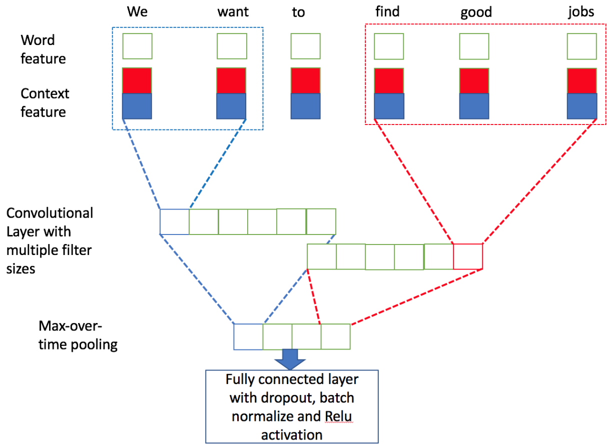
We
want
to
find
good
jobs
Word feature
Context feature
Convolutional Layer with multiple filter sizes
Max-over-time pooling
Fully connected layer with dropout, batch normalize and Relu activation

Figure 2: CNN-extracted

4 Experiments

4.1 Dataset and prepossessing

There are over 8million items of job information in our training data. Fitful prepossessing and feature engineering will help model convergence.

4.1.1 Text data prepossessing

We fix the number of words for each portion as described in Table 1, and pad the sentences whose lengths are shorter than the limit size.

Table 1: Different text max length

Title	Description	Work time	Requirement
22	410	36	43

4.1.2 Salary data prepossessing

Normalization method has a huge influence over numeric data. Considering evaluation convenient, we choose min-max normalization to transforms salary feature by scaling each feature to a given range. Some jobs are on hourly/daily wages basis, for convenience, we all convert them into annual income.

4.2 Experiment Settings

We propose the data as follows: We do not remove any stop words, symbols and emoji in the text, only splitting 10% of the dataset into a testing dataset and remaining 90% as the training set. Then we split 10% again from the training set into the validation set and the 90% as the real training set. We use "$Xavier$" and "He" initialization function and choose "*Relu*" as our activation function.

4.3 ResNet and Model parameters

ResNet is a useful method to learn different features in deep learning model of image classification task [5]. We also try to add a ResNet from input layer to CNN layer to add raw feature into CNN layer. Number of model parameters reflect different model complexity. Table 2 shows our experiment model parameters.

5 Results

We implement a series of models for our salary prediction task. Table 3 shows those models' prediction performance. We use Mean Absolute Error(MAE) as evaluation standard in our experiments. Since most of the annual income of Japanese jobs is under 6 million Japanese Yen, we set 6 million Yen, 8 million 10 million Yen upper limit and no upper limit, four kinds of max annual income range to evaluate our models. To the best of our knowledge, our Bid-GRU-CNN is the only model that surpassed the TextCNN. In particular, our model results show that the Bid-GRU-CNN approaches outperform and robust on our task. It proves that the Bid-GRU structure can effectively compose the semantic representation of words and strong robust of noise data than Bid-LSTM. Besides, CNN structure can capture more contextual information of features compared to TextCNN, and can avoid a certain degree vanishing gradient problem than RCNN.

5.1 Discussion

In majority of the past works, LSTM and GRU have close results on some tasks. LSTM even has better performance on some tasks because of its three gates structure. But in our experiments, the performance of GRU is much better than LSTM in terms of both accuracy and speed. This may be attributed to the following: There are two possible reasons cause our results.

- Our dataset includes some noise like emoji. For this kind of noise, GRU is better robust than LSTM.
- LSTM and GRU are just for word level representation in our model, the final feature learning work is done by CNN. LSTM three forget gate provide less information than GRU.

ResNet bad performance may be attributed to the two hypotheses, viz.

- Connect raw data from input layer, which also take lot of noises into CNN layer.
- Since ResNet is designed for getting different size features in CV task, it is not fit for our model and data.

6 Conclusion

Even though, the NLP general model is already very powerful on most of the tasks, but considering limited time and device in real service, data based deep learning model is rather a more practical way. Exploring more possibility on text regression with combination model is an important direction in both research and application development.

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[2] Jason P. C. Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. 2015. cite arxiv:1511.08308.

[3] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014.

[4] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image

Table 2: Numbers of model parameters

Model	**Total**	**Trainable**	**Non-trainable**
TextCNN	54,782,746	20,759,764	34,022,982
Bid-RCNN	37,854,953	3,824,905	34,030,048
Bid-LSTM-CNN	40,662,761	6,632,713	34,030,048
Bid-GRU-CNN	37,568,745	3,538,697	34,030,048
Bid-GRU-ResNet-CNN	53,234,665	19,208,201	34,026,464

Table 3: Salary prediction result by different Upper

Model	**6,000,000**	**8,000,000**	**10,000,000**	**All**	Train Time
TextCNN	383,425	423,973	453,842	493,802	1h
Bi-RCNN	416,527	460,755	492,457	551,886	10h
Bi-LSTM-CNN	385,719	425,627	452,860	508,968	18h
Bi-GRU-CNN	331,227	367,349	388,573	437,934	14h
Bi-GRU-ResNet-CNN	384,110	426,509	451,353	507,851	20h

recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[7] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. 08 2015.

[8] Shaun Jackman and Graham Reid. Predicting job salaries from text descriptions. 2013.

[9] Yoon Kim. Convolutional neural networks for sentence classification. 2014.

[10] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. pages 2267–2273, 2015.

[11] Yijun Zhou Luoxiao Li, Xutong Liu. Predicting of salary in uk. 2016.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[13] Tapio Nummi and Janne Salonen. Modeling and predicting individual salaries: A study of finland's unique dataset. 2005.

[14] Zhiyao Duan Rui Lu. Bidirectional gru for sound event detection. 2017.

[15] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997.

[16] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. 2016.