Deep Learning Model for Function Point Based Software Cost Estimation

—An Industry Case Study

Ming Qin, Lianteng Shen, Dahua Zhang
Information and Communication Department
Electric Power Research Institute Co. ,Ltd
Beijing, China

Liujun Zhao
State Grid Energy Research Institute Co. ,Ltd
Beijing, China

***Abstract*—Cost estimation of software engineering is a vital mission for software development organization. Accurate estimation can be valuable for both the developers and the customers. How to make estimation more accurate and reproducible is an important task to be settled in both research and industry area. This paper gives a review of the state of the art in the field of software cost estimation and proposes a Deep Neural Network(DNN) model to facilitate the usage of function point estimation. The results show that the function point based estimation can be a good choice in the early evaluation of the software budget while the deep neural network can significantly improve the labor productivity and the accuracy of the evaluation result .**

Keywords—deep neural network, software cost estimation, function point

I. Introduction

With the requirement from different industries, the scale of the software increased exponentially, to manage the software development, three important dimensions have been pay great attention on, such as efficiency, quality and cost, while the first two dimension have been studied a lot, cost estimation have been pay litter attention compare other two dimensions.

Cost has become one of the important issues in software development. Cost measurement for software helps a project manager to better analyze the engineering feasibility and plan the development efforts. Therefore, the requirement of the effective and efficient cost estimation methods has become a hot area[1].

Software cost estimation techniques could be grossly divided into 2 groups: specialist based estimation and model based estimation [2]. Specialist Judgment is one of most traditional technique [3] where the estimation of software cost is implemented based on expertise of parallel engineering. The advantage of this method is that this method takes valuable experience of expert, which can suitable for more complex scenario and more general topic however, it depends too much on the expert, which lacks standardization and also can be subjective. Lots of researches try to settle this question by construct different estimation mode where the cost of software is measured by utilizing varieties of engineering characters. COCOMO is first proposed in 1981 by Boehm, Barry W [4]. Lots of model have been present to estimate the cost of software more precise and can be more easy to use in different phrase of the software lifecycle, such as SLOC [5]. It can be used after the line of code have been predefined, IFPUG method [6], MKII method[7], NESMA method[8] and COSMIC method[9] are the typical function point based cost estimation model, which can be used in almost every phrase in the software development.

The core idea of these methods is to decompose the software development process into several separate units which may be the function point or the line of code. However, with the development of the cost estimation research, LOC have been argued for many reasons, such as it cannot be used before the software have been developed and also LOC cannot be used directly to measure the software cost without taking the complexity of the code into account. Contrarily, function point is convenient and can be widely applied to different fields. Now it has become a dominant model for software cost estimation, which have been developed into many standards by ISO organization. To make the FP analysis process more efficiency some useful tools have also been developed.

Function Point (FP) based cost estimation has been proved useful and also achieved success in many industry. However, it still exist some common tough challenges. One of them is its steep learning curve, the rule to model the estimation from requirement is remarkable complex, the FP analyst need to thoroughly read, remember and completely understand the rules defined in the guideline book which is nearly thousand pages. Moreover, the analysts need to practice over 50,000 function points to master the rule and apply them into the practice. The other is the process to distill the standard unit from requirement or design document. In large scale information system, the requirement or the design document can easily over thousand pages which fill with terminology and logic, it is not so easy for an analyst to understand the whole terminology and judge according to rule mentioned before.

In this paper, a method which combine the Conditional Random Field(CRF) and the Recurrent Neural Network(RNN) technique is proposed to mitigate the challenge mentioned above, both of these techniques provide robust results while the combination of these techniques can be more practicable for the further work. Fundamentally, the requirement is written in a nature language, from which we need to extract the entity that correlated to the FP, the output of RNN can be the segment of word, but the word can be composed into a term, which represents that the FP is not deterministic, so we put a CRF lay on the RNN to reduce the wrong prediction of sequence.

0-7695-6728-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICICAS48597.2019.00165

The rest of the paper is organized as follows. First, section 2 reviews the related work and Section 3 describes the Problem to be solved. Then, the proposed method is elaborated in Section 4 and Section 5 shows and explains the experimental results. Finally, the conclusion is summarized in Section 6 .

II. Related Works

Lots Neural networks(NN) based works have been dived into the topic of software cost estimations. Literature [1] take advantage of traditional feed forward neural network to measure software reliability. The multi-layer perception (MLP) and radial basis function neural networks are utilized separately in [10] and [11] for software effort estimation. A back propagation(BP) algorithm based neural network model is utilized in literature [12]. wavelet neural networks is applied in [13] for predicting software development cost. Fuzzy c-means clustering based radial basis function method is used to construct polynomial neural networks in [14]. [15] designs a hybrid framework which take advantage of both neural network and c-means method to measure software cost. The non-linear characters and complex relationship are ubiquitous in software cost measurement filed, which urge researchers to utilize complicated graph model for obtaining precise cost estimation.

III. Problem Definition

Challenges raised in FP analysis have been described in many research and industry practice, there are two most obvious barriers to conduct FP analysis, and the first is the rule to define the definition of different type of function point. The rule in different standard give out different method to classify the function point, moreover, the definition is described in text, which can be ambiguity between different domains and depend on the understanding of different analyst. The second is the workload to conduct a thorough function point analysis. In many cases, the requirement document is very detailed, so the entity and the relationship between the entities can be more complex and obscure, so if the requirement document can be work completely and done correctly is an important quality consideration for both the supplier and consumer and thus to the analyst.

Therefore, we can conclude these problems into 2 research questions.

RQ1: how to learn the rule from the labeled function point analysis document?

In industry, lots of function point analysis work have been done, and the rule to figure out these function point is almost the same. Therefore, if we can extract the patterns from the labeled document, the rules could be used for analyze the requirement in a smart way.

RQ2: Is there any learning method which is suitable for solving the software cost method effectively and efficiently?

Machine learning (ML) is a general concept for the static based algorithm to model the indeterministic feature and entity, however, the there are lots of ML algorithm to solve different type of ML task, so which kind of these method will be more suitable for function point analysis and in which level it can be helpful to improve the efficiency and quality of function point analysis is another important question we need to answer.

IV. Proposed method

In this section, our approach is proposed for automatic cost estimation via function point recognition. Figure 1 shows the basic structure of our method. The BiLSTM-CRF structure is firstly present to obtain target expressions from the requirement document.Then, the sentences in the document are classified into different type of function points.

The target expression extraction and function point classification are implemented by BiLSTM-CRF in this paper. In this process, named entity recognition (NER) framework in natural language processing field is referenced to solve the problem of function point recognition. In NER, a sentence is considered as a sequence of tokens which is marked with IOB(Inside, Outside and Beginning) tagging. A example with IOB tagging is shown in Table 1.

TABLE I. A sentence example with IOB tagging

Word	is	a	computer	game	software
Tag	O	O	B-EIF	I-EIF	I-EIF

Deep neural sequence model has shown promising success in NER [16], sequence tagging [17] and fine-grained opinion analysis [18]. An effective deep neural sequence model for NER is BiLSTM-CRF, which is combined with a bidirectional long short-term memory (BiLSTM) network as middle layer [19] and a conditional random fields (CRF) as output layer. Figure 2 shows the structure of BiLSTM-CRF, where the BiLSTM contains both forward and backward long short term memory networks to learn from precedings and tokens. Since the long short-term memory is a variant of recurrent neural network(RNN), it necessarily to introduce RNN, LSTM, BiLSTM and BiLSTM-CRF separately.

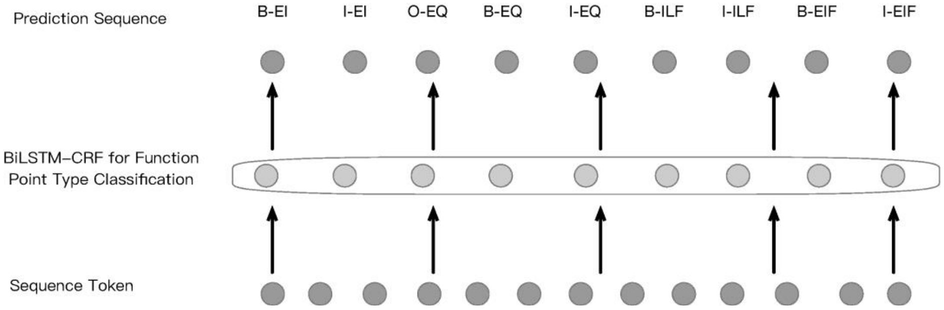

Fig. 1. Deep Artificial Neural Network based FP Recognition

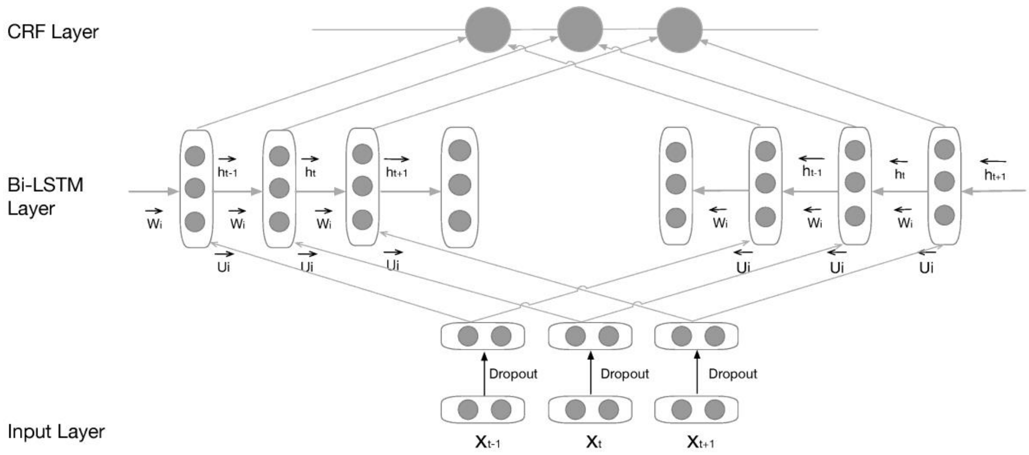

Fig. 2. Bi-LSTM CRF

RNN[22], where connections between nodes form a directed graph along a temporal sequence, is a typical NN for solving sequence problem. RNN could use its internal state (hidden unit h) to process the sequences of inputs $x = \{x_1, \Lambda, x_T\}$. The hidden unit h_t is updated dynamically based on the current input x_t and the previous hidden state h_{t-1} , where t is time step. The update process of h_t can be shown as follows:

$$h_t = g(Ux_t + Wh_{t-1}) \tag{1}$$

where U and W are update matrices of current input nodes and previous hidden nodes respectively; g is an predefined activation function,which is usually non-linear. With weight matrix V for output y_t ,the output of RNN can be obtained by equation $y_t = \mathrm{softmax}(vh_t)$, where h_t is the current hidden node and softmax is a frequently-used activation function.

In order to solve the problem of vanishing gradients in deep neural network, LSTM is designed with one more hidden state c_t which is also named cell state to reserve long term state. The conventional hidden state h_t which reserves short term state and the new hidden state c_t together reserve the long short-term information.

LSTM learns only from one direction,

The LSTM used in the BiLSTM-CRF [16] has two gates (an input gate i_t , an output gate o_t) and a cell activation vectors c_t .

BiLSTM uses two LSTMs to learn each token of the sequence based on both the past and the future context of the token. As shown in Fig.1, one LSTM processes the sequence from left to right, the other one from right to left. At each time step t , a hidden forward layer with hidden unit function $\overset{\rho}{h}$ is computed based on the previous hidden state $\overset{\rho}{h}_{t-1}$ and the input at the current step x_t and a hidden backward layer with hidden unit function $\overset{\upsilon}{h}$ is computed based on the future hidden state $\overset{\upsilon}{h}_{t+1}$ and the input at the current step x_t .The forward and backward context representations, generated by $\overset{\rho}{h}_t$ and $\overset{\upsilon}{h}_t$ respectively, are concatenated into a long vector. The combined outputs are the predictions of teacher-given target signals.

As another widely used sequence model, conditional random fields (CRF) is a type of discriminative undirected probabilistic graphical model, which represents a single log-linear distributions over structured outputs as a function of a particular observation input sequence.

Given observations variables X whose values are observed, random variables Y whose values the task requires the model to predict, and an undirected graph G where Y are connected by undirected edges indicating dependencies. CRF defines the conditional probability of a set of output values $y \in Y$ given a set of input values $x \in X$ to be proportional to the product of potential functions on cliques of the graph, shown in Equation (2).

$$p(y \mid x) = \frac{1}{Z_x} \prod_{s \in S(y,x)} \Phi_S(y_s, x_s) \tag{2}$$

Where, Z_x is a normalization factor overall output values $S(y,x)$ is the set of cliques of G , $\Phi_S(y_s, x_s)$ is the clique potential on clique s .

Afterwards, in the BiLSTM-CRF model, a softmax over all possible tag sequences yields a probability for the sequence y . The prediction of the output sequence is computed as Equation (3).

$$y^* = argmax_{y \in Y} \sigma(X, y) \tag{3}$$

Where, $\sigma(X, y)$ is the score function defined as Equation (4).

$$\sigma(X, y) = \sum_{i=0}^{n} A_{y_i, yi+1} + \sum_{i=1}^{n} P_{i, y_i} \tag{4}$$

Where, A is a matrix of transition scores, $A_{y_i,y_{i+1}}$ represents the score of a transition from the tag y_i to y_{i+1}. n is the length of a sentence, P is the matrix of scores output by the BiLSTM network, P_{i,y_i} is the score of the $y_i^t h$ tag of the i th word in a sentence.

As shown in Figure 2, dropout technique is used after the input layer of BiLSTM-CRF to reduce overfitting on the training data. This technique is firstly introduced by [23] for preventing complex co-adaptations on the training data. It has given big improvements on many tasks.

After target extraction by BiLSTM-CRF, all opinionated sentences are classified into EI,EO,EQ,EIF,ILF, according to the number of targets extracted from them.

V. Industry Case Study in State Grid

State Gird Ltd. is now one of the largest electricity transmission system operators in china. To support the more efficient operation of the business and the organization, lots of information system have been design and conduct to promote the information exchange and data storage. As the problem meet by other industry, to make the information system construct in a high quality, the control of expense in the software development life cycle is an important issue of both the budget sector and the software supplier group. with the development of the industry standard and the application of the function method, the work of function point recognition become more and more important, however, as we mentioned above, it is difficult for the company to train an efficient function point analyst in a short time, and also the accurate of the result can be subjective, so how to make this procedure more repeatable and more efficient is a crucial important issue from industry.

In order to fully utilize the experience of the function point analyst, a deep learning based function point recognition framework is described in Fig.3, the model we used here is described in Section 4, and we train this model by label the historical data and adjust the parameter of the deep learning model.

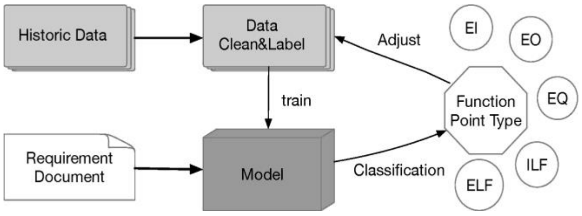

Fig. 3. Deep learning Based Function Point Recognition Framework

As shown in Figure 3, the historic data is first be cleaned and labeled during the data extraction procedure, and then be mapped into a word vector within a 100 dimensions space, after that we take these vectors as input to the BI-LSTM model, and train the BI-LSTM model to predict the type of the word, after the word have been predict, the CRF model is used to model the sequence in the final model version.

We applied this frame work in 52 project while separate these project into train, validate and test group in a ratio of 6:1:3, we applied this model in keras with tensorflow backend, which is an open source implementation of deep neural network.

TABLE II. Train Result

Layer Type	Output Shape	Param#
Embedding	(none,none,200)	228600
Bidirection	(none,none,200)	240800
CRF	(none,none,11)	2354

In this paper, neural network was trained on the 471,754 words which extract from the industry requirement document, and organized the 5871 samples and validate on 2977 samples, from the output of the training process we can illustrate that, the model include the BiLSTM and CRF layer, the output dimension is 200 and connected neural unit is also 200 with fore-propagation 100 and back-propagation 100. The validate result is represented in the Table.3.

As shown in Table.3, the sample number of ELF is much less than the other function point type, and the result also shows the overfitting of the prediction of ELF type. However, the other type of function point get a sound performance compare to the state-of-art work, the average accurateness achieve about 70%, and it can be seen that the number of sample can be an important factor, which may has great impact on the performance of the model.

TABLE III. Train and Validate Result

FP type	Test num	Train num	accurateness
EI	406	617	65.8%
EO	1407	1756	80.1%
EQ	55	103	53.3%
ILF	224	293	76.4%
ELF	693	16	18%

VI. Conclusion

A novel approach is proposed in this paper to improve function point analysis task in software development. The proposed method utilizes BiLSTM-CRF framework to model and discriminate function point in sentence-level, which takes full advantage of time sequence information and continuity information. Experiments shows that the proposed method achieves state-of-the-art on the field of function point analysis. In the future, the problem of lacking of enough samples shall be solved to avoid under fitting of the training process.

References

[1] A. Smith and A.K. Mason. Cost estimation predictive modeling: Regression versus neural network. The Engineering Economist, vol. 42, pp. 137-161, 1997.

[2] E. Mendes, N. Mosley, and I. Watson. A comparison of casebased reasoning approaches. In Proceedings of the 11th international conference on World Wide Web, pp. 272-280. ACM, 2002.

[3] R. T. Hughes. Expert judgement as an estimating method. Information and software technology, vol. 38(2),pp.67-75, 1996.

[4] B.W. Boehm. Software engineering economics, vol. 197. Prentice-hall Englewood Cliffs (NJ), 1981.

[5] A. J. Albrecht and J. E. Gaffney. Software function, source lines of code, and development effort prediction: a software science validation. IEEE transactions on software engineering, vol.6, pp.639-648, 1983.

[6] Function Point CPM IFPUG. Release. 4.1. IFPUG, Westerville, OH, 1999.

[7] C. R. Symons. Software sizing and estimating: Mk II FPA (function point analysis). John Wiley & Sons, Inc., 1991.

[8] F. Niessink and H. V. Vliet. Predicting maintenance effort with function points. In Software Maintenance, 1997. Proceedings., International Conference on, pp. 32-39. IEEE, 1997.

[9] R. Meli, A. Abran, V. Ho, and S. Oligny. On the applicability of cosmic-ffp for measuring software throughout its life cycle. In Proceedings of the 11th European Software Control and Metrics Conference, pp. 18-20, 2000.

[10] G. E. Wittig and G. Finnic. Using artificial neural networks and function points to estimate 4gl software development effort. Australasian Journal of Information Systems, 1994.

[11] N. Tadayon. Neural network approach for software cost estimation. In Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, vol. 2, pp. 815-818. IEEE, 2005.

[12] A. Idri, A. Zakrani, and A. Zahi. Design of radial basis function neural networks for software effort estimation. IJCSI International Journal of Computer Science Issues, 2010.

[13] K. Kumar, V. Ravi, M. Carr, and N. Kiran. Software development cost estimation using wavelet neural networks. Journal of Systems and Software, vol.81(11), pp.1853-1867, 2008.

[14] S. Roh, S. Oh, and W. Pedrycz. Design of fuzzy radial basis function-based polynomial neural networks. Fuzzy sets and systems, vol.185(1),pp.15-37, 2011.

[15] V. Khatibi, D. Jawawi, S. Hashim, and E. Khatibi. A new fuzzy clustering based method to increase the accuracy of software development effort estimation. World Applied Sciences Journal, vol.14(9),pp.1265-1275, 2011.

[16] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360, 2016.

[17] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[18] O. Irsoy and C. Cardie. Deep recursive neural networks for compositionality in language. In Advances in neural information processing systems, pp. 2096-2104, 2014.

[19] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on, pp. 6645-6649, 2013.

[20] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[21] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, vol.9(8), pp.1735-1780, 1997.

[22] M. ndez-Gavilanes, T. lvarez-Lopez, J. Juncal-Martınez, E. Costa-Montenegro, and F. Gonzalez-Castano. Unsupervised method for sentiment analysis in online texts. Expert Systems with Applications, vol.58, pp.57-75, 2016.

[23] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.