2015 Seventh International Conference on Computational Intelligence, Modelling and Simulation

A Deep Learning Approach for Categorizing Risk Impact in Software Domain

Baala Mithra SM
Global Technology Office
Cognizant Technology Solutions
Chennai, India
baalamithra.sm@cognizant.com

Kuhelee Roy
Global Technology Office Cognizant Technology Solutions
Chennai, India
Kuhelee.Roy@cognizant.com

Sanglap Sarkar
Global Technology Office
Cognizant Technology Solutions
Chennai, India
sanglap.sarkar@cognizant.com

Venkateshwar Rao Madasu
Global Technology Office
Cognizant Technology Solutions
Chennai, India
Venkateshwar.madasu@cognizant.com

Subrahmanya VRK Rao
Global Technology Office
Cognizant Technology Solutions
Chennai, India
Subrahmanyavrk.rao@cognizant.com

Raj Bala
Global Technology Office
Cognizant Technology Solutions
Chennai, India
Raj.Bala@cognizant.com

Abstract - **This paper addresses the problem of identifying the impact areas of risk from a given text description about the risk. The challenge of this piece of work lies in the fact that the description is in natural language. Literature provides a wide range of proposed framework where statistical machine learning techniques have been used to predict the risk from quantitative features. This work views the problem in a natural language processing perspective. In order to envisage a more accurate classification of the risk impact category we have used a deep learning paradigm.**

Keywords - n-grams, convolution, deep learning, backpropagation, bag of words

I. INTRODUCTION

In software domain, the task of risk prediction requires analysis of historical data pertaining to similar projects for assessing two vital things viz, estimating the probability that the objectives of the project will be reached and that the objectives have been actually reached when certain risk have occurred. According to [1], a vital component of a risk management process is identifying and analyzing the risk data. The risk data, pertaining to a software project development usually contains attributes like category, exposure, stage, impact area etc. [2]. Traditional machine learning techniques have been efficiently used in combination with evaluation metrics like Neural Network and Support Vector Machine in order to compute the prior and posterior probabilities for the failure and success of the project. The influx of modern technology necessitates rapid progress in software development to support the same.

Identifying risks and the related impact areas requires analyzing the historical data along with the risk stage, exposure and status. Finding the relationship between the attributes characterizing a risk is a critical part in analyzing historical data related to risk in software domain. Studies on impact of risk factors in large-scale IT projects have been provided in [3] [4]. A list of current software risk items has been provided in [5]. The main idea behind analyzing historical data related to risk and predicting the impact of risk in future is to aid in anticipating and avoiding problems prior to their occurrence. [6]

Unlike other works in literature, this work sees the problem of predicting impact of risk, from a natural language processing perspective. The challenge lies in the fact that the traditional POS tagging and chunking techniques for processing sentences in natural language will not suffice for the current problem. In order to attain greater accuracy deep learning methodology has been used.

II. RELATED WORK

Few existing framework for risk management have been provided in [7-9]. According to [10], budget, schedule, technical qualities are the important factors used to evaluate if the project objectives are met. On the other hand, according to [11], the factor contributing to the success of risk analysis depends on the way a risk is described. In [1], metrics like Domain, KSLOC, and Complexity has been used in order to obtain the impact areas of risk. The issue of delay risk has been dealt with in [12]. Machine Learning techniques have been used for learning the relationship existing among various attribute characterizing a risk [1] [2]. In [13], the chances of unforeseen circumstances related to failure or damage in terms of monetary aspects has been dealt with.

Prioritizing the risk and classifying the impact into high, medium and low was also considered as a part of risk history analysis. Classifying risks from low to high including questionnaire was highlighted in [14].

Though fair accuracy rates have been achieved in using machine learning approaches for analyzing history of risk and their impact, a better approach encapsulating the semantic features as well as other metrics is still an area of concern. Unlike shallow learning approaches where the features are learned explicitly and classified, deep learning

2166-8523/15 $31.00 © 2015 IEEE
DOI 10.1109/CIMSim.2015.31

deals with learning the features implicitly towards classification. Various deep learning architectures include deep neural networks [15], convolutional deep neural networks [16], deep belief networks [17], Restricted Boltzmann Machine (RBM) [18].

III. PROBLEM STATEMENT

A. *Risk prediction*

Given:

1. risk description (RD)
2. Risk category (RC)
3. Risk exposure (RE)
4. Risk status (RSS)
5. Risk stage (RST)
6. Impact area (RIA)

Expected outcome: Predicting the risk impact area.

The following nine categories of RIA(s) have been considered in this paper:

1. Cost
2. Legal
3. Quality
4. Timeline
5. Reputation
6. Budget Overrun
7. Future Revenue
8. Financial Penalties
9. Business Disruption

B. *Shallow learning for classifying risk categories*

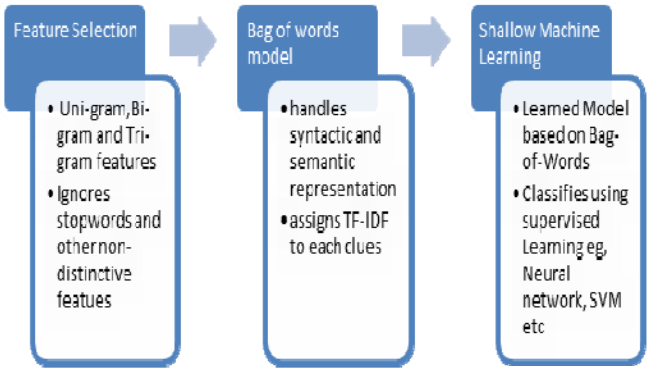

Figure 1. Shallow Learning framework for risk prediction

A. *Deep learning for modelling risk description*

As, mentioned in literature identification of risk at a higher accuracy rate is indispensable for the sake of attaining project completion objectives. Since the input for risk categorization is in natural language our proposed method attempts to use the deep learning model for Natural Language Processing as proposed by [15].

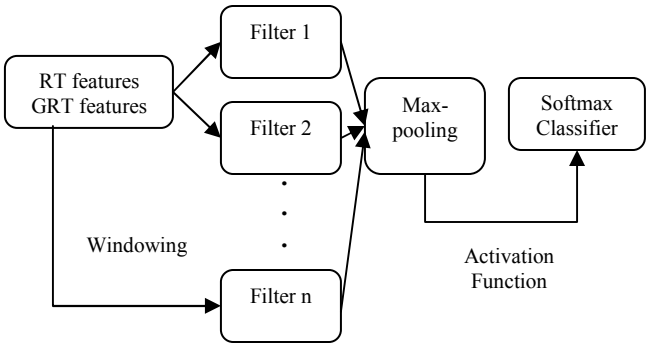

Figure 2. Deep Learning framework for risk prediction

The main objective is to classify the RIA (Risk Impact Area) from RD (Risk Description) and RC (Risk Category). The input to the first stage of the model used in this work is given by the concatenation of RD and RC. Since RD contains words that are common for more than one category risk, including the RC feature narrows down the feature characterizing a risk of specific type, thereby defining a distinctive feature clues for identifying the RIA.

The challenges of this work are as follows:

1. Unlike using numerical attributes, this work deals with input that is written in natural language.
2. The risk description written in natural language does not follow any specific syntax. Also the sentences might not be complete in terms of parts of speech

IV. METHODOLOGY

A. *Identifying Risk Token(RT)*

First, the uni-gram, bi-gram and tri-gram are transformed into a Bag-Of-Words vector. This is followed by learning the sentence-level features using convolutional approach. The method used in this paper has been explored by Collobert et al. (2011), in the context of POS tagging, chunking (CHUNK), Named Entity Recognition, (NER) and Semantic Role Labeling (SRL). WordNet which is a database containing synonyms, holonyms, and hypernyms around 150,000 words is used while populating the database of keywords having distinctive feature, characterizing a risk. Other words are mapped to a set of special words that might characterize a wide range of risk types.

B. *Identifying Group of Risk Token(GRT)*

The conventional lexical features include noun sequences. For a risk description such pairs has been considered for each bi-gram and tri-gram. The list of those pairs is also considered to be a part of dictionary used for training. For example, for a bi-gram denoted as:

(RT1-RT2)

The following lexical features are considered:

- Left and right token of RT1
- Left and right token of RT2

1) Paragraph Level Features

The single word token as well as the pair of token limits the representation of the semantic composition of a risk description. The reason behind this is that pairs of token always might not suffice to represent the discriminating feature of a particular risk type by putting together long distance tokens within a paragraph. Paragraph describing a risk contains sentences. Windowing is applied on the entire paragraph without the consideration of stopwords separating two sentences. This leads to window feature (WF) and position feature (PF). This technique is similar to the one explored in [15]. Word features (WF) results in a vector of concatenated partitions of tokens in sentence/(s). For example for a sentence with 6 words, each word starting from index 0 as given below:

S: [w0 w1 w2 w3 w4 w5 w6]

The WF with window size 3 would result in a vector as follows:

{[wo,w1,w2], [w1,w2,w3], [w2,w3,w4], [w3,w4,w5] , [w4,w5,w6]}

While WF considers the entire paragraph, PF limits itself to one sentence only. The distance of the word (if contributes to positive value for BoW vector) is computed as follows:

Let the dictionary of uni-gram contains words: w_1, w_2, w3, w4…. w_n

Let the sentence contains words: s_1, s_2, s3, s4…. s_n

For word

$s \in \{s_1,....s_n\}$, that also belongs to the dictionary of uni-gram, the contribution of the word s is considered to be positive. Let such words in a particular sentence be denoted by

$s_{p1}, s_{p2},...., s_{pn}$

For each of the words belonging to $s_{p1}, s_{p2},...., s_{pn}$, the position feature is denoted as the distance between the first and the last word of the sentence the word belongs to. This contributes to the emphasis of the particular word based on its position in the sentence.

2) Feature concatenation by Convolution

A set of local features computed around words that contribute to the histogram of Bag-Of-Words. Each local feature is in turn the result of the windowing approach. Our next step is to apply a convolution function on the outputs of the window approach.

One-dimensional convolution is computed by dot product of the weight vector m with each of the outputs of the windowing approach. The type of convolution used is of type narrow that yields a vector of size smaller than the original ones.

C. Convolutional Deep Learning of GRT

The convolutions results in the following operation:

a) Dot product of weight vector with the output of the window processing stage:

For a sentence s and a weight vector m1, the one-dimensional convolution with each m-gram in the sentence results in another sequence denoted:

$$c_j = W_1^T s_{j-(m-1)+1:j}$$

Where,

- j is the token number in the sentence
- m is the size of the filter (depends on varying window sizes eg, 3,4,5)

eg, for the sentence in the description of risk of category *Contractual Definition and Tracking*,

First round of UAT is completed.

The windowing process with m=3 and j=4 *(UAT)* gives the following results,

Token j: 'of'

j-(m-1)+1gives 3. (negative indices are ignored)

'of UAT is'

- $s \in \Re^{n_0 \times t}$,
  - $n_0 = m * n$ *(n is the dimension of feature vector, depends on the number of tokens in the sentence)*
- $W_1 \in \Re^{n_1 \times n_0}$
  - n_1 *is the size if the hidden layer for each* n_0 *(with varying window sizes)*
- *The resultant feature map can be represented as:* $c = [c1, c2 \ldots c_{n-m+1}]$
- *This operation results in a vector where the values correspond to different filters.*

b) Dealing with varying sentence size:

In order to deal with varying sentence lengths, the maximum of each row in the matrix c is taken, denoted by $c_{\max}$. This represents the most useful feature in a particular dimension and is independent of the size of a sentence, which is further related to the size of a paragraph. This corresponds to the most distinctive feature corresponding to a particular filter.

c) Selection of activation function:

Hyperbolic tanh function is selected as the activation function:

$$\frac{d}{dx}\tanh x = 1 - \tanh^2 x$$

This function can be used for the backpropagation training stage. Hence the non-linear transformation in this stage can be written as:

$$g = \tanh(W2c_{\max})$$

Where, $W2 \in \Re^{n_2 \times n_1}$

n_2 is the number of hidden layer 2

d) Final Layer

This output is fed as input to a softmax classifier that outputs the probability score for each of the category labels.

e) Training

The backpropagation network needs to learn the weights W1 and W2.

V. Experimental Results

A. Data Description

TABLE I. Training and Testing Data

Training Data		Testing Data	
Category 1	42	Category 1	20
Category 2	31	Category 2	18
Category 3	49	Category 3	17
Category 4	37	Category 4	22
Category 5	38	Category 5	15
Category 6	44	Category 6	12
Category 7	43	Category 7	20
Category 8	48	Category 8	16
Category 9	39	Category 9	22

B. Confusion Matrix using Shallow Learning

TABLE II. Classification of Risk Impact Area using Neural Network

Predicted										Error
Cat 1	7	3	10	0	0	0	0	0	0	0.65
Cat 2	0	0	18	0	0	0	0	0	0	1
Cat 3	0	0	14	0	3	0	0	0	0	0.17
Cat 4	0	0	3	19	0	0	0	0	0	0.13
Cat 5	0	0	0	5	10	0	0	0	0	0.33
Cat 6	0	0	1	0	0	11	0	0	0	0,083
Cat 7	0	0	0	12	0	0	8	0	0	0.6
Cat 8	0	0	0	4	0	0	10	2	0	0.373
Cat 9	0	0	4	0	0	0	0	0	18	0.18

C. Confusion Matrix using Deep Learning

TABLE III. Classification of Risk Impact Area using Deep Net

Predicted										Error
Cat 1	14	0	6	0	0	0	0	0	0	0.3
Cat 2	0	16	2	0	0	0	0	0	0	0.11
Cat 3	0	0	14	0	3	0	0	0	0	0.176
Cat 4	0	0	1	21	0	0	0	0	0	0.45
Cat 5	0	0	0	2	13	0	0	0	0	0.133
Cat 6	0	0	1	0	0	12	0	0	0	0
Cat 7	0	0	0	6	0	0	14	0	0	0.3
Cat 8	0	0	0	6	0	0	10	13	0	0.187
Cat 9	0	0	4	0	0	0	0	0	20	0.091

Learning the risk description using the deep learning approach shows error rate that is comparatively lesser to that of using shallow learning approach.

VI. Conclusion

This paper addresses the challenges existing in the current methods of risk classification in terms of two aspects viz, features related to risk and shallow learning approaches. The contribution of this paper can be summarized as follows:

1. Unlike using numerical metrics for classifying risks, this paper deals with features that are written in natural language.
2. Furthermore, since the descriptions written are not complete in terms of parts of speech, extraction of features corresponding to each risk type plays an important role.
3. The last section provides comparative results in terms of error rate with respect to shallow learning and deep learning technique.

References

[1] Sarcià, S. A.; Cantone, G. & Basili, V. R. (2007), A Statistical Neural Network Framework for Risk Management Process - From the Proposal to its Preliminary Validation for Efficiency., *in* Joaquim Filipe; Boris Shishkov & Markus Helfert, ed., 'ICSOFT (SE)' , INSTICC Press, , pp. 168-177

[2] Thitima Christiansen, Pongpisit Wuttidittachotti, Somchai Prakancharoen and Sakda Arj-ong Vallipakorn, 'Prediction Of Risk Factors of software development project by using multiple logistic regression', ARPN Journal of Engineering and Applied Sciences, 2015

[3] B. Michael, S. Blumberg, and J. Laartz, "Delivering large-scale IT projects on time, on budget, and on value," 2012. [Online]. Available: http://www.mckinsey.com/insights/businessn technology/deliveringn large-scalen itn projectsn onn timen onn budgetn andn onn value

[4] S. Group, "Chaos report," West Yarmouth, Massachusetts: Standish Group, Tech. Rep., 2004.

[5] B. W. Boehm, "Software risk management: principles and practices," Software, IEEE, vol. 8, no. 1, pp. 32–41, 1991.

[6] Ying, Q., Meng-Jia, Y., and Feng, L. (2012). The Risk Factor Analysis For Software Project Based on the Interpretative Structural Modelling Method. International Conference on Machine Learning and Cybernetics.

[7] Madachy, R.J., 1997. Heuristic Risk Assessment Using Cost Factors. *Software*, pp. 51-59, IEEE CS Press.

[8] Roy, G. G., 2004. A Risk management Framework for Software Engineering Practice. *ASWEC04, Australian Software Engineering Conference*. IEEE CS Press.

[9] Alberts, C.J., 2006. Common Element of Risk. *TN-014*, pp. 1-26, CMU/SEI.

[10] Standish Group, 1994. CHAOS Report 1994-99. http://www.standishgroup.com, last access Feb. 2006.

[11] Jones, C., 2002. Patterns of large software systems: failure and success. *Computer,* N.o 23, pp. 86-87, IEEE CS Press.

[12] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran and Aditya Ghose, Characterization and prediction of issue-related risks in software projects, Proceedings of 12th Working Conference on Mining Software Repositories (MSR), co-located with ICSE 2015, IEEE (acceptance rate 30%). To Appear. (ACM SIGSOFT Distinguished Paper Award)

[13] Tim, M., Osamu, M., Yasunari, T., and Tohru, K. (2009). Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects. J. Software Engineering & Applications

[14] Koolmanojwong, S. (2010). The Incremental Commitment Spiral Model Process Patterns for Rapid-Fielding Projects.PhD Dissertation, Department of Computer Science, University of Southern California.

[15] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: International conference on Machine Learning, ACM, 2008, pp. 160-167.

[16] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classication with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.

[17] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: International Conference on Machine Learning, ACM,2009, pp. 609{616.

[18] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted boltzmann machines for collaborative filtering, in: International Conference on Machine Learning, ACM, 2007, pp. 791-798