Journal of Applied Computer Science & Artificial Intelligence

Research Paper Vol.-2, Issue- 1, December 2021 ISSN: 2582-8673

Differential Evolution based Parameter Optimization on Stochastic Gradient Descent Learning for Bio-molecular Event Trigger Extraction

Amit Majumder[1*]

[1] JIS College of Engineering, email:cseamit49@gmail.com

Available online at: http://jacsai.org/

Received: ……./2021, Revised: ……..2021, Accepted: ……2021, Published: 30/June/2021

***Abstract*—** Event extraction deals with finding more detailed biological phenomenon, which is more relatively challenging compared to simple binary relation extraction like protein-protein interaction. In this paper we present a differential evolution-based optimization technique in order to determine the most optimized parameters in a machine learning framework which is then used to extract event triggers from biomedical text. Event trigger is a part of an event expression. Event trigger detection step corresponds to the identification of triggers and classifies them into predefined nine categories of interest using a multiclass classifier. We use Stochastic Gradient Descent (SGD) learning, which is trained with a diverse set of features that cover both statistical and linguistic characteristics. Experiments on the benchmark datasets of BioNLP-2011 shared task datasets show the recall, precision and f-score values of 69.92%, 78.61% and 73.89% respectively for event trigger detection.

***Keywords*—** Event extraction, Trigger detection, Dependency graph, SGD learning, Parameter Optimization

I. INTRODUCTION

Biomedical documents in electronic form are growing rapidly in the Inter- net. There is a recent trend for fine-grained information extraction from the text [7], which was addressed in consecutive text mining challenges [4, 5, 8, 6].

In this paper we propose parameter optimization (PO) technique for event trigger detection. The event triggers are classified into 9 potential types. Among these, five are simple which corresponds to gene expression, transcription, protein catabolism, phosphorylation and localization. The rest four events, namely binding, regulation, positive regulation and negative regulation are relatively complex. We use classification algorithm based on SGD learning. To optimize the parameters of this algorithm, we have used Differential Evolution (DE) algorithm [11]. The system is evaluated using BioNLP 2011 shared task datasets. Evaluation results show the state-of-the-art performance on this benchmark datasets.

II. MOTIVATION

To extract event trigger from bio-medical data, we need to consider several features like content feature, contextual feature, syntactic features. All these features are text features. Before applying learning algorithm, these text features need to be converted numerical feature and after conversion feature dimension becomes huge (more than 5 lacs). This numerical representation of text features is in sparse matrix form. To run DE, it needs some iterations to get optimized result. To find out fitness of one candidate in a generation, it needs to apply the learning algorithm on these high-dimensional features. If there are P number of candidates (i.e., population) in one generation, then it needs to run the learning algorithm P number of times in that iteration. Therefore, for many generations, it needs to run the learning algorithm several times. As SGD classifier is very fast and suitable for more than 10^5 training examples and more than 10^5 features, we use this learning algorithm. The default parameter values provided in implementation of the algorithm do not generate good result. So, we apply parameter optimization technique and for optimization we choose Differential Evolution (DE)

[*]***Corresponding Author: Amit Majumder***

III. MAJOR STEPS FOR EVENT TRIGGER EXTRACTION

In this section we describe our major steps for event trigger extraction. For trigger detection, we use supervised classifier using SGD learning[1]. Parameters of this learning algorithm are optimized using differential evolution algorithm [12]. Major steps for event trigger detection are given below.

A. *Segmentation and Tokenization*

Tokenization and segmentation of text are done using Genia tagger[2]. After tokenization, we remove the sentences which do not have proteins, as triggers are applied on to proteins.

B. *Feature Extraction*

Different types of features are needed to identify the trigger words. These features include contextual, semantic and syntactic features which are extracted from the dataset.

C. *Optimization of Parameters using DE*

For event trigger detection, we use SGDCLASSIFIER[3], a classifier using SGD learning. This algorithm has certain parameters to be learnt. Parameters can be both numerical or categorical. Numerical parameters are *alpha, epsilon* etc., whereas categorical parameters are *loss function, penalty, learning rate* etc. We apply Differential Evolution (DE) to optimize these parameters. These parameters are used to encode the candidate solution. We represent all the parameters by numerical values. Representing candidate solutions having numerical values are straightforward, but for representing categorical parameters, it needs to be converted to numerical values.

The conversion process from parameters to candidate for DE is illustratedin figure-3

Algorithm 1: Candidate creation for DE:

candidate_NPS=[NP_0,NP_1,...,NP_{m-1}] /* Here, *NPS* indicates Numerical Parameters; each NP_i is a numerical parameter*/
candidate_CPS=[CP_0,CP_1,...,CP_{n-1}] /* Here, *CPS* indicates Categorical Parameters; each CP_i is a list of possible options */
Inputs:
m = Number of numerical parameters;
n = Number of categorical parameters ;
[low, high]=bound of a parameter; /* lower bound and upper bound of parameter */
Outputs: candidate
Begin

1. candidate_NPS=[] /* initially empty list */
2. For i=0 to m-1:
   i. r=random number within [low, high] /* [low, high] is the bound of NP_i */
   ii. Append r in the list candidate_NPS
3. candidate_CPS=[] /* initially empty list */
4. For i=0 to n-1:
   i. r=random number within [low, high] /* [low, high] is the bound of CP_i */
   ii. Append r in the list candidate_CPS
5. candidate= candidate_NPS + candidate_CPS /* Merging */

End

Figure 1: Candidate creation for DE

Algorithm 2: Converting numerical form of candidate_CPS to categorical form during SGD learning

Inputs: candidate_CPS
Outputs: category_form

Begin

1. For i=0 to n-1: /* n is number of categorical parameters */
   (a) value=candidate_CPS[i]
   (b) options=CP_i
   (c) k=size of options /* CP_i is a list of k options i.e. $CP_{i,j}$ for j=0 to k-1 */
   (d) index=Integer(value * k)
   (e) category_form=$CP_{i,index}$

End

Figure 2: Converting numerical form of candidate CPS to categorical form during SGD learning

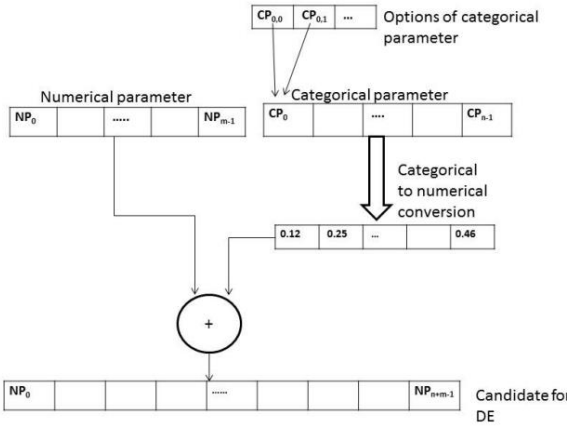

Figure 3: Formation of Candidate used by DE

D. *Fitness Computation for DE*

For the fitness computation, the following steps are executed.

1) Suppose, there are N number of parameters (categorical and non-categorical both) in a particular candidate

2) Construct a classifier with SGDCLASSIFIER on training dataset using these parameters.

3) Apply the classifier on development dataset. compute recall, precision andf-score.

4) f-score value is used as fitness of the candidate for the objective function. The objective is to maximize this objective function using the search capability of DE.

IV. Experimental Results and Analysis

We use the BioNLP-ST-2011 datasets and evaluation frameworks for the experiments. We perform parameter optimization technique on the development dataset, and use the configurations obtained for final evaluation. We perform 5- fold cross-validation. Statistics of BioNLP-11 dataset for genia event extraction has been mentioned in table 1.

Attributes	Training	Development	Test
Abstracts +Full articles	908 (5)	259 (5)	347 (5)
Sentences	8,759	2,954	3,437
Proteins	11,625	4,690	5,301
Total events	10,287	3,243	4,457

Table 1: Statistics of BioNLP-ST 2011 Genia Event dataset (training, development and test). Value inside parentheses indicates the number of full articles

We apply parameter optimization technique using DE. We have used SGD- CLASSIFIER, which learns using SGD technique and it supports SVM or LR algorithm based on categorical parameters. The following parameter values have been set for DE:

- Population size=70
- Number of generations=20
- Mutation=0.225
- Strategy to create trial vector= best1bin
- Probability of crossover (i.e., recombination) =0.9.

In the implementation of SGDCLASSIFIER, it has 19 parameters. Out of these 19 parameters we choose 7 parameters which mostly have effect on performance of classifier. All these parameters are defined in implementation of SGDCLASSIFIER, implemented in scikit-learn[8]. After running DE for parameter optimization using SGDCLASSIFIER on Bio-NLP 2011 genia dataset, the optimized parameters that we have obtained are shown in Table-2. The result with parameter optimization is given in table-3 and shown in the form of confusion matrix in figure-4.

Parameter	Default Value	Optimized Value
alpha	0.0001	2.13e-05
epsilon	0.1	2.06
power t	0.5	0.26
loss	hinge	hinge
penalty	l2	elasticnet
learning rate	optimal	invscaling
class weight	None	None

Table 2: Default parameter and optimized parameter values for SGDCLASSIFIER

Class	Recall	Precision	F-score
neg	98.94	97.4	98.17
Gene expression	80.04	85.05	82.47
Localization	74.42	66.67	70.33
Phosphorylation	90.28	85.53	87.84
Transcription	48.72	90.48	63.33
Protein catabolism	90.48	95.0	92.68
Binding	69.65	84.83	76.5
Regulation	46.41	64.67	54.04
Positive regulation	51.56	75.93	61.42
Negative regulation	51.79	73.42	60.73
Average of All	96.59	96.22	96.3
Average of Trigger Classes	61.71	78.72	68.67

Table 3: Trigger detection without Parameter

Class	Recall	Precision	F-score
neg	98.76	97.92	98.34
Gene expression	81.56	86.67	84.04
Localization	76.74	76.74	76.74
Phosphorylation	87.5	87.5	87.5
Transcription	58.97	83.13	69.0
Protein catabolism	95.24	95.24	95.24
Binding	74.32	86.43	79.92
Regulation	53.11	64.53	58.27
Positive regulation	67.31	72.95	70.02
Negative regulation	61.61	75.82	67.98
Average of All	96.94	96.7	96.79
Average of Trigger Classes	69.92	78.61	73.89

Table 4: Trigger detection using Parameter Optimization by DE

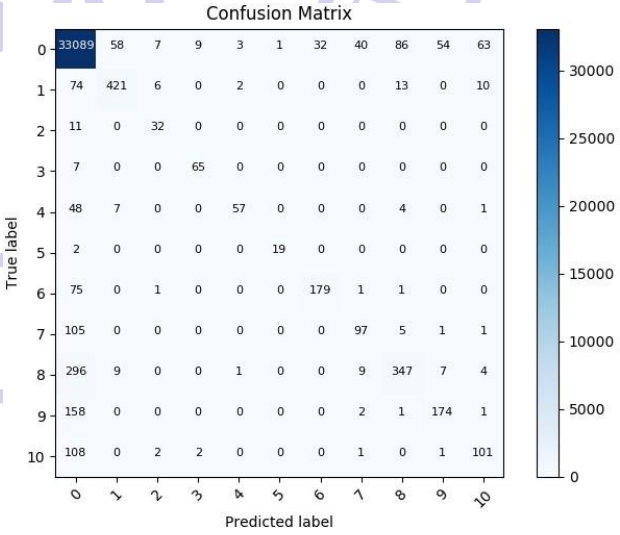

Figure 4: Confusion Matrix: Trigger detection without Parameter Optimization; 0: None (i.e., Not a trigger or an entity), 1: Gene

ISSN: 2582-8673

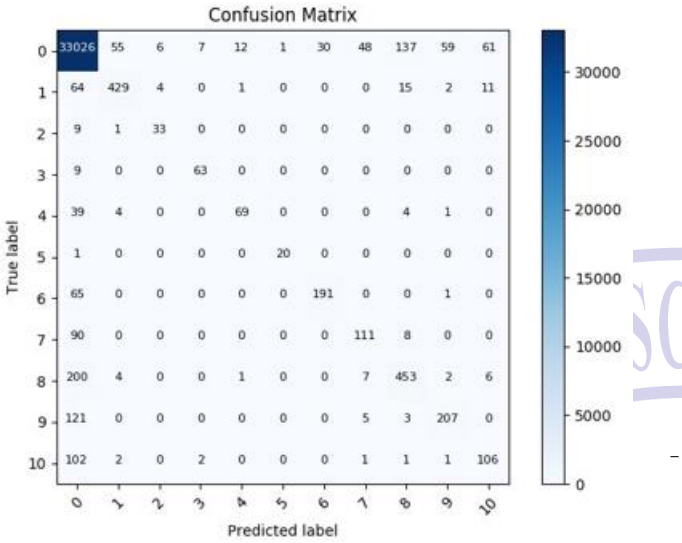

Figure 5: Confusion Matrix: Trigger detection with Parameter Optimization by DE; 0: None (i.e. Not a trigger or an entity), 1: Gene expression, 2:Localization, 3:Phosphorylation, 4:Transcription, 5:Protein catabolism, 6:Binding, 7:Regulation, 8:Positive regulation, 9:Negative regulation, 10:Others (i.e. entity)

From experimental results, we find that average f-score value of the technique without parameter optimization is 68.67%, whereas with parameter optimization technique we achieve 73.89% f-score. Therefore, we achieve nearly 5 points increment in f-score using parameter optimization technique. If we analyse the results for individual classes, we see that for some classes the proposed technique significantly improves the performance. Improvement of f-score values for *Gene expression, Localization, Transcription, Protein catabolism, Binding, Regulation, Positive regulation and Negative regulation* are approximately 2%, 6%, 6%, 3%, 3%, 4%, 9% and 8%, respectively using parameter optimization technique.

A. Comparison with existing systems

Trigger detection is first step to extract genia event expression. Event extraction systems generate results for event expression, not for trigger detection. So, it is not possible to compare our experimental results with the results of event extraction systems. We compare our results with the existing results on trigger detection. Existing result for trigger detection on BioNLP 2011 dataset is 67.3% [14] f-score, whereas our trigger detection system shows 73.89 %f-score (+6.5%). Therefore, our trigger detection system performs better than other trigger detection systems.

B. Error Analysis

In order to gain more insights, we analyse the outputs of our proposed technique to find the errors and their possible causes. From the confusion matrix as shown in figure-5, we find that in development dataset there are total 526 *Gene expression* type triggers, out of which 429 have been identified correctly. Some tokens have been identified as *Gene expression type* incorrectly. There are 55 tokens which originally belong to *None type* words, have been identified as *Gene expression* type triggers. This is the token which originally belongs to the *Localization* type, but our system detects the token as *Gene expression* type trigger. There are 4 tokens which originally belong to *Tran- scription* type, but they have been identified as *Gene expression* type triggers. Also, we find that 4 tokens belonging to *Positive regulation* have been identified as *Gene expression* type triggers. Out of 43 *Localization* type triggers, 33 number of triggers have been correctly detected. There is total 72 triggers as *Phosphorylation* type, out of which 63 have been identified correctly. Total 9 tokens have been identified as *Phosphorylation* type trigger incorrectly. From the result it is clear that for *Gene expression, Localization, Phosphorylation, Transcription, Protein catabolism and Binding type* triggers error is less as compared with other type of triggers. This happens because, these triggers are applied on proteins only, whereas other category of triggers *(i.e., Regulation, Positive regulation and Negative regulation)* are regulatory triggers which are complex type triggers. As event trigger detection is a part of overall event expression extraction, we cannot compare these experimental results with other event extraction systems already available. Event extraction systems measure performance based on event expressions, not on trigger detection. In this paper, we have measured the effect of parameter optimization techniques and we see that this technique highly improves the performance of system. As trigger detection is one step of event expression extraction, error generated in trigger detection will propagate to event expression extraction. As error is reduced using parameter optimization technique, propagation of error will be less for eventexpression extraction.

V. CONCLUSION AND FUTURE WORKS

In this paper we propose an efficient technique for event trigger extraction based on parameter optimization by DE using SGD as learning algorithm. Overall performance of the system by parameter optimization technique is re- call, precision and f-score values of 69.92%, 78.61%, and 73.89% respectively, whereas the technique without PO shows recall, precision and f-score values of 61.71%, 78.72% and 68.67% respectively. Therefore, PO technique shows an improvement of 5% in f-score value. Overall evaluation results suggest that there is still the room for further improvement. In our

future work, we would also like to apply deep learning approach to check whether performance of trigger extraction improves or not. We would also like to find out the arguments of these identified triggers and to find event expression.

REFERENCES

[1] [CHINCHOR, N. Message understanding conference (muc-7) proceedings. In Overview of MUC-7/MET-2 (1998).

[2] D., M. Any domain parsing: Automatic domain adaptation for parsing. Ph.D. Thesis (2010).

[3] DAVID MCCLOSKY, M. S., AND MANNING, C. D. Event extraction as dependency parsing for bionlp 2011. In Proceedings of BioNLP Shared Task 2011 Workshop (June 2011), pp. 41–45.

[4] HYOUNG-GYU LEE,HAN-CHEOL CHO, M.-J. K. J.-Y. L. G. H. H.- C. R. A multi-phase approach to biomedical event extraction. in bionlp09. In Proceedings of the Workshop on BioNLP, pp. 107–110.

[5] JIN-DONG KIM, SAMPO PYYSALO, T. O.-R. B. N. N. J. T. Overview of bionlp shared task 2011. In Proceedings of BioNLP Shared Task 2011 Workshop (June 2011), p. 16.

[6] KIM,J.-D., WANG,Y.,COLIC,N.,BEAK,S. H.,KIM,Y. H., ANDSONG, M. Refactoring the genia event extraction shared task toward a general framework for ie-driven kb development. In Proceedings of the 4th BioNLP Shared Task Workshop (2016), pp. 23–31.

[7] KIM J-D, OHTA T, P. S.-K. Y. T. J. Overview of bionlp09 shared task on event extraction. In BioNLP 09: Proceedings of the Workshop on BioNLP (2009), pp. 1–9.

[8] LISHUANG LI, YIWEN WANG, D. H. Improving feature-based biomedical event extraction system by integrating argument information. In Proceed- ings of the BioNLP Shared Task 2013 Workshop (August 2013), p. 109115.

[9] LYNETTE HIRSCHMAN, M. K., AND VALENCIA, A. Proceedings of the sec- ond biocreative challenge evaluation workshop. In CNIO Centro Nacional de Investigaciones Oncologicas.

[10] NEDELLEC, C. Learning language in logic -genic interaction extraction challenge. In Proceedings of the 4th Learning Language in Logic Workshop (LLL05) (2005), pp. 31–37.

[11] ONWUBOLU, G. C., AND DAVENDRA, D. Differential Evolution: A Hand- book for Global Permutation-Based Combinatorial Optimization, 1st ed. Springer Publishing Company, Incorporated, 2009.

[12] STORN, R., AND PRICE, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11, 4 (1997), 341–359.

[13] VOORHEES, E. Overview of trec 2007. In Sixteenth Text REtrieval Con- ference (TREC 2007) Proceedings.

[14] WANG JIAN, WU YU, L. H.-F.-Y. Z.-H. Biological event trigger word extraction based on deep syntactic parsing. Computer Engineering (2014), 25–30.