



# Biomedical event extraction on input text corpora using combination technique based capsule network

R N DEVENDRA KUMAR<sup>1,\*</sup>, K SRIHARI<sup>2</sup>, C ARVIND<sup>3</sup> and WATTANA VIRIYASITAVAT<sup>4</sup>

<sup>1</sup>Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore, India

<sup>2</sup>Department of CSE, SNS College of Technology, Coimbatore, India

<sup>3</sup>Department of ECE, Karpagam College of Engineering, Coimbatore, India

<sup>4</sup>Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok, Thailand

e-mail: devapsna@gmail.com; harionto@gmail.com; arvichakra@gmail.com; hardgolf@gmail.com

MS received 6 October 2021; revised 27 July 2022; accepted 19 August 2022

**Abstract.** Biomedical Event Extraction (BEE) is a demanding and prominent technology that attracts the researchers and scientists in the field of natural language processing (NLP). The conventional method relies mostly on external NLP packages and manual designed features, where the features engineering is complex and large. In addition, the conventional methods on BEE uses a pipeline process that splits a task into many sub-tasks, however, the relationship between these sub-tasks is not defined. In this paper, such limitations are avoided using the combination technique that relies on Capsule Network (CapsNet) to perform a task. The CapsNet is used for the extraction of feature representation from the input corpora and then the combination technique reconstructs the events from RNN output. This method extracts the tasks from a BEE over several annotated corpora that extract the events from the molecular level in case of multi-level events. The proposed model is compared with state-of-the-art models over various text corpora datasets. The results show an improved rate of accuracy of CapsNet classification over cancer biomedical events than the existing methods.

**Keywords.** Biomedical event extraction; biomedical triggers; machine learning; CapsNet.

## 1. Introduction

PubMed in 2020 recorded more than 30 million papers [1] on biomedical literature. Text mining has evolved to be a common research area and a great demand for technology as the results provided in biomedical literature attract a great deal of interest and the vast amount of literature remains a challenge to obtaining knowledge; Biomedical Event Extraction (BEE) is a basic technique of text mining that is an efficient means of portraying organized information from unstructured text [2]. The dynamic and arbitrary nature of events in biomedicine makes the extraction of events a complicated process, so it is necessary to conduct research in connection with this [3].

Based on the concept of Biomedical event [4] which consists of an event-type trigger term and multiple arguments according to BioNLP. In this, an argument has a relationship between an event trigger and another event. Hence, to understand the event source, the function of event extraction is required.

Since extraction of biomedical events is a standard activity, where different methods are suggested by the

researchers to support the BEE. In most of the previous works, three styles are possible and are followed which include rule-based methods [5, 6], where it contains the standard models of low-level or deep learning.

Deep learning approach is preferred to handle task-specific case models which are rarely used in BEE. Deep learning models are focused on the tasks of event elimination, for event trigger identification [7–9] and classification of the relationship between the events [10–13], where most of which produces higher accuracy of detection than the conventional approaches. The conventional models typically have two drawbacks considering the effectiveness of existing technologies for retrieval of biomedical events. Firstly, the models depend mostly on manual features and typically involve the dynamic Natural Language Processing (NLP) of generalized NLP toolkits. Secondly, these approaches arrange and split the task into subtasks in a pipeline way that simplifies the issue but it further lacks the relationship between the subtasks and renders the mechanism suspended for the build-up of mistakes.

The present work deals with a surge of basic operations in convolution before the capsule which reduces the high

\*For correspondence

dimensional data and encodes the several functionalities with various kernel sizes. The parallel convolution layer cascade is implemented by kernel sizes comparatively greater than [14, 15] so larger kernel sizes can be used for integrating more details on large receptive fields. While multi-task education was used [16] to improve training data by exchanging information between tasks, in addition to its normal use of encoding text in vectors, we have introduced a much cheaper alternative to increase our data sizes using the Word2vec model. While it is promising to merge CapsNets and RNNs in [17, 18], the additional use of deep learning can confuse them. Moreover, as CapsNets can grab contexts and well encrypt sequences, it is of minor significance to achieve the advantages by integrating CapsNet. It can be concluded that the model should be sufficiently versatile to adjust changes, such as altering the order of terms, because of the high uncertainty of the sentence design. This flexibility was part of the reason why it is preferred for static routing over dynamic routing. Even if this argument is often true, the method argues that the original order of the word forms an important aspect in the general sense of a sentence because of modification of the words order that typically leads to a meaningless sentence.

CapsNet is more robust and efficient for feature representation. In CapsNet, the utilisation of feature extractor helps in mapping the feature vector into matrix form during the extraction of features. The CapsNet uses a Combination Strategy (CS) to fuse the spatial relationships using feature matrix that forms a 3-D information cube. To mitigate the complexity while finding the optimal information cube combinations, CapsNet uses a 3-D convolutional kernel that helps in the construction of features and an encoder finds simultaneously the spatial relationships and features.

In this paper, we propose a novel multi-level event extraction model over the multi-biomedical event with a Capsule Network (CapsNet) model with a combination strategy (CS). The CapsNet detects the event triggers from the raw input corpora. The CS is used to construct the suitable events from the detected triggers, which is obtained from the CapsNet feature extraction. The CS is used for the integration of the data vectors and it determines the event set extracted. The method does not consider the CS for training and it is used directly for validation. The CapsNet model avoids the need for feature engineering and hence the features obtained from the word embedding are combined in semantic space. The CS reduces the classification error rate while forming a suitable relationship with the event triggers.

The major contributions of the paper are given below:

- The CapsNet is utilised for feature extraction and combination strategy is used to combine the events from the feature detected in order of finding suitably the event triggers with the suitable relationship.
- The authors developed an event-building modelling using CS that integrates the vectors of extracted data to determine the event set getting extracted.

- The authors evaluate the model with biomedical event tasks like Cancer Genetics (CG), Multi-Level Event Extraction (MLEE) and BioNLP Shared Task 2013 (BioNLPST2013).
- The model is tested against the state-of-the-art techniques to estimate the level of accuracy than the conventional methods.

The outline of the paper is given below. Section 2 provides the related and existing works. Section 3 discusses the proposed model. Section 4 evaluates the results and discussion with state-of-the-art models. Section 5 provides conclusion with the possible directions for future scope.

## 2. Related works

A wide variety of analyses are conducted on the study of the sentence classification. Previous approaches to non-neural networks focused on the topical classification and attempting to filter documents based on their subject. Applications such as market intelligence or another product evaluation, however, require a more advanced and comprehensive examination of the viewpoints raised than a topical classification.

TEES [19, 20] is a BEE method, using rich dependence parsing features. The TEES uses a multi-class SVM step-by-step approach, which split-up the entire task into simple and consecutive graph node/edge classification tasks.

EventMine [21] is a technique on hand-crafted feature extraction model using SVM pipeline system. Majumder *et al* [22] used the biomedical case extraction stacking model. The two types of classifiers are basic and meta-level classifiers where SVC, SGD and LR are used as the basic classifier and SVC is meta-level classifier. Another method using a beam search that uses a hierarchical perception for encoding and decoding is a transformation-based paradigm for event extraction [23] to find a universal projection.

In the present era, Deep learning approaches are implemented to improve textual representation and improve the rate of efficiency. Wang *et al* [24] proposed a method to create a multi-distributed convolutional neural network (CNN) for the extraction of BEE. In addition to the embedding, the distributed features often cause forms, POS labels, and subject representation. In order to remove biomedical occurrences Li *et al* [25] used dependence-based term embedded and parallel multi-level CNN. This method offers additional details by grouping a multi-segment expression separated by word and argument.

In order to include the additional functionality, Björne and Salakoski [26] merged the CNN with the original TEES and replaced the SVM classification with dense layers indicating a major improvement in efficiency by the incorporation of the neural network. Li *et al* [27] suggested a system for the extraction of biotopes and bacterial events

using gated, where the repeated unit networks with an activation function.

Pang *et al* [28] focused on linguistic heuristics or a pre-selected collection of words. In order to conduct sentence classification on movie review datasets, Naïve Bayes classification is used. The problem is that they need a previous awareness or chosen seed terms.

The use of rules-based classifiers is another common technique for the classification of sentences, specifically to perform query classification. For Text REtrieval Conference (TREC) classification problems, Silva *et al* [29] use the rule-based classification device with the Support Vector Machine (SVM). While these methods were able to record advanced results, the use of manually designed patterns renders it unfit for functional applications particularly where large and complex data sets are available.

Hermann *et al* [30] have been using combinatorial category grammar operators to derive semantic relevant representations from sentences and phrases of variable size, and each vector representation is given as an Autoencoder for the classification function.

The automated classification of text sentiments in Dufourq *et al* [31] was also using genetic algorithms (GAs). They also suggested a GA approach to classify the phrase by unknown terms to be either an emotion or an enhancing term which may surpass those structures in their work, referred by the writers as a genetic sentimental analysis algorithm (GASA).

Kim [32] suggests a superficial CNN network with only single convolution and pooling layers for penalty classification. The investigator contrasted his performance with a Word2vec pre-trained (CNN static), random (CNN-rand) and a fine-tuned Word2vec embedding (CNN non-static) of regular datasets. For most datasets used in the experiment the recorded findings indicate the superior precision. In addition, the author emphasizes that, as a common purpose, a pre-trained Word2vec model can be used for text embedding.

A CNN with a complex k-max combined which can control phrases of various lengths significant for sentence modelling was adopted by Cheng *et al* [33]. The CNN for best optimisation Kousik *et al* [34] and digital emotion analyses containing texts, images has also been embraced by Cai *et al* [35]. Zhang *et al* [36] used a CNN architecture for the complete interpretation of text from inputs at character-level. Conneau *et al* [37] established the powerful CNN architecture that indicates that the efficiency of this model is improved with the depth. Although most CNN text classification models are not very profound. While these CNN models provided substantial results, it has demonstrated a strong functioning CapsNet model.

An RCNN model for text classification that generated strong results in integrating a standard RNN structure and max pooling by CNNs was presented by Lai *et al* [38]. A repetitive layout, a two-way RNN with word embedding, is

used in both the directions to catch contexts and the output is spread over a full layer of pooling.

Cheng *et al* [33] has developed a long-term short-term memory network for memory-and-awareness logic. Instead of a single memory cell that makes adaptive memory used in the replication of neural attention, the LSTM architecture was expanded with a memory network.

Zhao *et al* [15] have implemented a series of capsules whose output is supplied to achieve the finished effects by the capsule average layer of pooling. During their work, by using Leaky-Softmax and the coefficient of modification they modified the routing by agreement algorithm given in [39]. In accordance with prior CNN initiatives, the findings showed a substantial change.

Similarly, Kim *et al* [40] introduced an ELU-gate device before a convolution capsule by using a text classification with the Caps Net-based architecture. Both the dynamic routing and its simpler version were experimented by the authors as static routing. Their points were that simpler static routing yielded superior results in terms of classification precision and power to dynamic text classification routing.

CapsNets were developed by Xiao *et al* [41] for multi-task learning text classification. The exchange of information between tasks during routing functions to the relevant tasks was a difficulty in multi-task learning. The routing-by-agreement of CapsNets is capable to group a function which helps to resolve the issues encountered by inequality in multi-tasking learning for each task. An updated task routing algorithm has been proposed for this reason, so that choices are feasible between tasks. In order to substitute the algorithm for dynamic routing, CapsNets for text classification tasks were also introduced with RNN and its variants.

Similarly, Saurabh *et al* [42] designed an LSTM CapsNet for toxicity to be identified in statement. Saurbh *et al* [43] single module capsule network gives good results. CapsNet is more robust and efficient for feature representation [44–48].

Dhiman *et al* [49] presented a novel algorithm called Spotted Hyena Optimizer (SHO) which analysed the social relationship and collaborative behaviour of spotted hyenas.

The huddling behaviour of the Emperor Penguins was analysed by Dhiman *et al* [50] using the Emperor Penguin Optimization (EPO) algorithm.

Kaur *et al* [51] introduced the Tunicate Swarm Algorithm (TSA) which is a bio-inspired meta heuristic optimization algorithm. This proposed algorithm is similar to that of the swarm behaviours of tunicates and propulsion of jets. Another bio-inspired algorithm for solving constrained problems in industries was developed by Dhiman *et al* [52]. This algorithm evaluates the convergence behaviour and complexity in computational procedures.

A comparative study of the maximizing and modelling production costs using composite and triangular fuzzy and trapezoidal FLPP was studied by Kumar *et al* [53] and its

intense effects were investigated. Chatterjee [54] discussed the importance of AI and CRI. The huge impact caused by the Corona virus in Indian states have been analysed by Vaishnav *et al* [55] by utilizing various machine learning algorithms.

Gupta *et al* [56] provided an optimal suggestion using machine learning approaches for crime tracking in India. A deep learning model using convolutional neural networks with transfer learning to detect the accuracy of breast cancer was given by Sharma *et al* [57]. Shukla *et al* [58] introduced as novel approach to address various issues in performance encountered in Multicore systems.

### 3. Proposed method

In this paper, we propose an extraction model over multi-biomedical events with a deep neural network model with a combination technique. The former model is the CapsNet that detects the event triggers and its suitable relation from the raw input corpora. The latter technique utilises the detected event triggers and constructs the suitable events. CapsNet model eliminate the need of feature engineering and it ensures that the features are obtained from the word embedded in case of semantic space. The utilisation of combination technique helps to reduce the error rate while classifying to form a suitable relationship with the event triggers.

#### 3.1 Preprocessing

This section provides the details of simple pre-processing operations, where the input data is pre-processed before it is connected to the convolutional network. The input sentence from the dataset is split into words to replace the word by its corresponding vector representation. The analysis uses the process of tokenization for vector representation. A stripping technique or a string cleaning mechanism [1] is used as tokenizer for the process of tokenization. Once the tokenization is completed, data augmentation and word embedding takes place. The study uses Word2vec model [22] trained previously with the publicly available datasets, where this model is equipped with pre-trained 3 million phrases and words. The length of each sentence may vary based on the input datasets. To maintain constant length, the architecture of CapsNet is maintained with fixed dimensions for the given input. In this paper, we use zero padding to maintain the constant length of given input sentence.

#### 3.2 Sentence model

The input is fed with a sentence from the dataset, which is of the form

In Figure 2, the input stage is denoted by a sentence as given below

$$x_1 \oplus x_2 \oplus \dots \oplus x_n,$$

where

$x_i$  is the essential word and

$\oplus$  is the concatenation operation.

In this paper, we use 2D representation ( $n \times d$  dimensions) to define the sentences in the vector form. In this 2D representation, we define  $d$  as the embedding dimension (say 300) and  $n$  as the total number of words in a sentence. In the present case, each row is represented in the form of individual words as in Figure 1.

#### 3.3 CapsNet architecture

The architecture of CapsNet is illustrated in Figure 2. The structure of CapsNet [3] is usually built with a convolutional layer (first layer), primary capsules layer (second layer), and a digital capsules layer (third layer). The capsules are the group of neurons with different parameters on its activity vector for a specific entity or a feature [3]. The magnitude of a vector represents the probability of feature detection and its instantiation parameters represent the orientation. The output of capsules in CapsNet is sent to its preceding parent layer using a dynamic routing algorithm upon an agreement as in [3]. It further maintains the spatial information that offers resistance to attacks than the convolutional neural network.

Figures 2 and 3 used in the work involve the series and parallel convolution layers to test the efficacy of feature extraction. The results show that the proposed method with parallel convolutional layer performs well than the series convolutional layer of figure 2.

The testing of CapsNet architecture with a single, two and three convolutional layers with/without parallel operations is given in Figure 3. Comparative results between the three architectures show an improved accuracy rate with the one given in Figure 2, where its accuracy is 4.5% greater than the other two architectures. On other hand, a parallel primary capsule layer is used after each

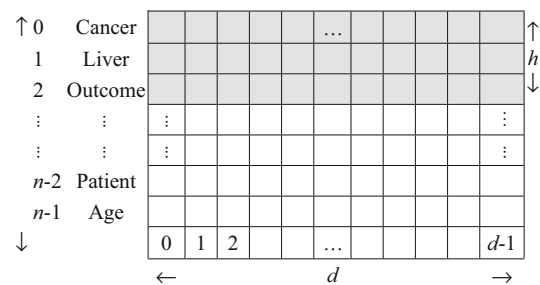
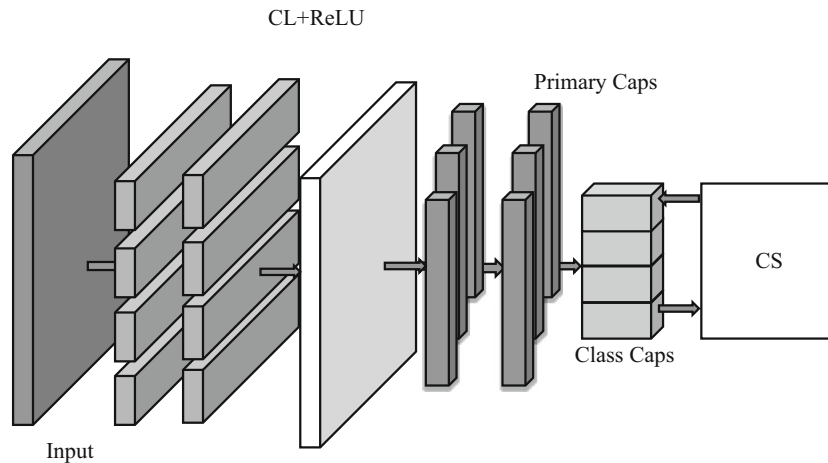
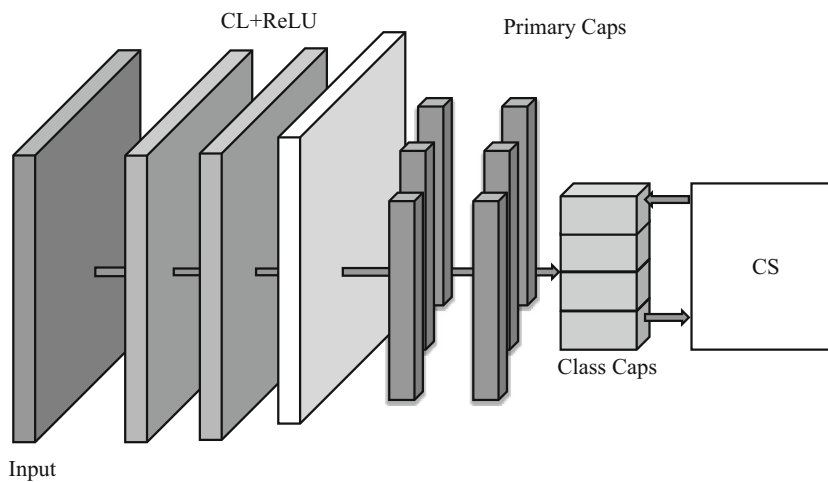


Figure 1. Sentence model.



**Figure 2.** Proposed CapsNet architecture for sentence classification.



**Figure 3.** CapsNet without parallel convolutional layer.

convolutional block, where the results are compared with the architecture as in Figure 2. The results of Figure 2 show no noteworthy difference with the other two architectures, therefore the CapsNet with least complexity architecture is given in Figure 2. It is seen that the initial stages contain a convolutional layer arrays, where various kernel sizes are used for the extraction of features.

3.3.a *Convolution Stage.* The convolution stage carries out a parallel operation for the extraction of input features from the input data corpora. The entire column is applied with convolutional operation as each single row contributes to a word i.e., sliding window of  $h \times d$  kernel size is used, which is given in Figure 1. In this case, the total number of words is represented as  $h$  is used in a single convolutional step. It hence offers two different advantages that it maintains sequential words order and reduces the total convolutional operations to process the input data. These two

advantages enable the reduction of total parameters that accounts for increased speed of CapsNet.

A feature  $f_i$  is generated for a given filter  $K^{h,d} \in \mathbb{R}^{h \times d}$  as:

$$f_i = \varphi \left( \sum_{h=1}^h \sum_{d=1}^d K_{h,d} X_{i+h,d} + b_i \right) \quad \text{for } i = 1, 2, \dots, n,$$

where

$f_i$  is defined as the features generated,

$\varphi$  is defined as the nonlinear activation function, while ReLU is used as an activation function,

$K^{h,d}$  is defined as the filter say  $m$  set,

$X_i$  is defined as the input word vector

$b_i$  defined as the bias term using convolution operation.

Hence, the formation of column vector ( $F$ ) occurs from the concatenation of features ( $f_i$ ) at the 1<sup>st</sup> convolutional stage.

From Figure 2, it can be seen that the parallel convolution operation utilises the different kernel size. It acquires the possible word combinations from the input corpora and these features are used for the process of classification.

At the second stage, the results of the operations of parallel convolutional are concatenated in order to form column vector ( $m$ ), which is equal to total number of filters used. The output of the second stage is then forwarded to the subsequent stages of CapsNet, where a dropout unit is placed between them i.e. it is placed between the 2<sup>nd</sup> convolutional layer and primary caps layer. The output data from the 2<sup>nd</sup> convolutional layer is then processed in the primary caps layer for the extraction of features, where the operations are made optimal by reducing the input data dimensionality by the convolutional stages. Such reduction of data dimensionality enables the CapsNet to perform its operations in an optimal manner.

**3.3.b. Primary Caps Layer.** The primary caps layer is the second convolution layer in CapsNet that tends to accept the scalar inputs from the first layer. The primary caps layer outputs the vector format of the scalar input, where it is defined as capsules that helps the preserve the semantic word representation and local order of words. This layer is an 8 dimensional with 32 filter channels, where the detection of features at the previous layer offers a good combination of selected features. These features are finally grouped in the form of capsules.

Henceforth the primary caps layer replaces the scalar input to vector output, since the upcoming higher or next level requires a dynamic routing agreement. The capsule or a group of vectors may get activated, even if multiple capsules tends to agree with a specific capsule in its next layer. This accounts for the propagation of results from the lower level i.e. convolutional stage to the higher level i.e., class caps layer. Therefore, a capsule cluster is formed with the activated cluster from its lower layer. It enables the higher level to produce a high probability output based on the presence of the entity and its high dimensional pose vector.

The dynamic routing mechanism offers the connection of Primary Caps Layer with its subsequent layer i.e., Class Caps Layer. If  $u_i$  is the single capsule output, then it produces a vector as below:

$$\hat{u}_{ji} = w_{ij} \cdot u_i$$

$w_{ij}$  is the translation matrix

$i$  is the capsule in the present layer and

$j$  is the capsule in the next or higher level.

A higher-level capsule is then fed as the lower level output sum, where the higher level represents the next layer i.e. output and the lower level represents the Primary Caps Layer. Here a coupling coefficient  $c_{ij}$  is then multiplied with output sum of lower level during the process of dynamic routing, which is represented as below:

$$s_i = \sum_j c_{ij} \cdot \hat{u}_{ji}$$

$c_{ij}$  is the coefficients and it is estimated using a softmax function, where it is defined as below:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

where

$b_{ij}$  is defined as the initial parameter, which is given as

$$b_{ij} = \hat{u}_{ji} v_j + b_{ij}.$$

Here,  $b_{ij}$  selects the probability of coupling between the capsule  $i$  and  $j$

$v_j$  is defined as the output vector within the [0, 1] and the expression is given with a squashing function as

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}$$

The vector length is also represented in terms of the probability of the presence of a capsule in that layer.

**3.3.c. Class Caps Layer.** The last layer in the CapsNet is the Class Caps, which accepts the values from primary caps layer based on dynamic routing agreement. This layer is a 16 dimensional per class with the routing value of 3 for its iterative routing process.

### 3.4 Combination strategy

The study obtains the final support vectors of features from the CapsNet, where these values are obtained in relation with relation, candidate trigger and event. The CS is finally used for the integration of the obtained data vectors and determines the extracted event set in case of classification. However, the method does not consider the CS for training and it is used directly for validation.

The CS aims to reduce the penalty score, since it measures the lack of consistency between the CapsNet output and final events from CS. The penalty score is defined with two different parts that includes the following:

- If a vector relation consists of a positive value and if it fails to appear in final extracted events: It is then referred as support waste, where the penalty score reports the support waste to the Class Caps Layer.
- If a vector relation consists of a negative value and if it appears in final extracted events: It is then referred as support lacking, where the penalty score reports the support lacking to the Class Caps Layer.

The support waste and support lacking have conflicting objectives that operates in contrast with each other. With increasing candidate events in the final set, minimal is the support waste penalty and higher is the support lacking. On the other hand, with reducing candidate events in the final set, minimal is the support lacking penalty and higher is the support waste. Hence, the feedback to the CapsNet layer i.e.

Class Caps Layer enables the reduction of total penalty in the final extracted events.

Finally, the target is given as below:

$$C_{best} = \arg \min_{c \subseteq C} score_{pen}(c)$$

where

$c_{best}$  is defined as the extracted event set,

$c$  is defined as the enumeration of all the subsets of  $C$ :

The penalty score is hence defined as below

$$score_{pen}(c) = \left( \sum_{event_k \in c} \max(1 - \alpha s_k^{(e)}, 0) + \beta \sum_{trigger_i \notin c} s_i^{(t)} + \gamma \sum_{trigger_i \notin c} s_i^{(r)} \right)$$

where

$s_k^{(e)}$  is the support of an event ( $k$ ),

$s_i^{(t)}$  is the support of a trigger ( $i$ ),

$s_i^{(r)}$  is the support of a relation ( $j$ ),

These penalty factors are finally reweighted using  $\alpha$ ,  $\beta$  and  $\gamma$  parameters in the CS.

The  $score_{pen}(c)$  is optimized by enumerating the subsets of  $C$  by including the exponential time complexity. In case of complex events from certain sentences, many candidate events may lead to complex computational cost. Hence, CS acts as an approximation strategy to resolve the  $O(n^2)$  time complexity problem.

The initial set of events chosen is an empty set and then the addition of candidate events are done in greedy manner. The candidate events are initially sorted in topological order and independent handling of the events in related with the different event trigger takes place. The handling of events is not considered for the support values if it exists between the trigger events. Once the candidate event set is received at CS, the support values of the vectors from the CapsNet including its candidate events, relations and triggers in a sentence. Thus, the final set of events or vectors extracted from the dataset is returned.

The nested events are a serious concern in BEE, where it is prevented by using loop detectors after the process of CS. This helps to avoid event loop formation. The addition of extracted event in the final set in an iterative manner enables the elimination of events present in the event loop. Therefore, the event modification is assigned on all events based on the modification vectors available in the final extracted set.

## 4. Results and discussions

The experiments are conducted on TensorFlow backend in Keras on an Intel Core i7 machine with 8700 CPU operating at 3.20 GHz with 32 GB RAM, where three GeForce GTX 1080 Ti acts as the GPUs.

### 4.1 Datasets

This section provides the discussion on training and evaluation of the model using three annotated datasets that includes MLEE, CG and BioNLPST2013. The datasets are split into training and testing, where 80% of data from all the three data is used for training and remaining 20% for testing.

### 4.2 Evaluation

The MLEE consists of events related to multi-level events at the molecular to the organ level. CG consists of events related to cancer that includes cellular tissues, molecular foundations and organ effects. Finally, PC consists of events related to the data triggers on biomolecular pathway models. The study considers only the datasets with entity labels for each individual word that may enable the task to focus on the extraction of target events. In this study, the pre-processing of documents is considered simple, since the document is split into sentences and finally it is tokenized into word sequences. Such pre-processing enables the reliability of the model on itself and not on any NLP toolkits.

The CapsNet model is simulated with the succeeding hyper parameters that includes the following:

- Batch size = 30,
- Learning rate = 0.001,
- Total epochs = 50 and
- Routing iteration = 3.

These parameters are chosen in such a way that it reduces the time complexity associated with the task of feature extraction and event selection.

### 4.3 Training of CapsNet

To improve the performance of the CapsNet-CS, the study conducts the experiment using a 2-phase training. In this training mode, the weights of CL are initialised and the Glorot uniform initializer initialises  $w_{ij}$  and the uniform coupling coefficients initialises  $b_{ij}$ . The study trains the network with three datasets and iterations are conducted to achieve optimal accuracy. Further, Glorot uniform initializer and uniform coupling coefficients re-initialises  $w_{ij}$  and  $b_{ij}$  by training the CapsNet by retaining the trained weights of CLs. More optimisation can be achieved if the CL gets the proper initial weights from Glorot Uniform Initializer and Uniform Coupling Coefficients. This produces optimal accuracy by learning the features from the input corpora.

From the Table 1, it is seen that the 2-phase CapsNet-CS obtains result on accuracy than other deep network models on all three datasets. This shows an effective understanding of data by the proposed model, where it recognises well the

**Table 1.** Precision on various datasets.

Network	CG	MLEE	BioNLPST2013
Convolution neural network	75.51	90.11	82.66
LSTM	75.91	90.26	84.20
2-phase CapsNet	76.32	90.29	84.21
Convolution neural network (static)	77.80	91.60	84.70
Convolution neural network (non-static)	79.80	92.20	86.20
Convolution neural network (multichannel)	81.00	92.80	86.40
Capsule-B	81.02	92.80	86.50
Capsule (static)	81.10	93.00	86.51
MCapsNet	81.50	93.60	86.80
SVMS	81.60	94.20	86.80
DCNN	82.10	94.50	86.80
LR-Bi-LSTM	82.30	94.80	87.20
CCAЕ	82.34	95.00	88.10
Bi-LSTM+SWN-Lex	83.50	96.00	88.60
BLSTM-2DCNN	83.80	96.10	89.20
CapsNet	84.53	96.46	89.50

features from the input corpora than the CNN and other deep learning models.

#### 4.4 Results on varying of Kernel sizes

This section presents the experimental results to observe the effects of varying kernel size on parallel layers in CapsNet-CS. The parametric comparison is collectively made in terms of Kernel size with 1st CL and 2nd CL. The performance of the 2-phase CapsNet-CS is compared with CapsNet-CS model. From Table 2, it is seen that the datasets with longer length of sentences produces better results, when its kernel size in CL is large.

- The 1<sup>st</sup> layer i.e. the CL have 4 parallel CL, where the selected kernel size of each CL is  $CL_1 = (7, 300)$ ,  $CL_2 = (8, 300)$ ,  $CL_3 = (9, 300)$  and  $CL_4 = (10, 300)$ . Each CL is embedded with 64 filter set and a stride of 1.
- The 2<sup>nd</sup> layer is again a CL with 4 parallel CL, where the selected kernel size of each CL is  $(CL_1, CL_2, CL_3, CL_4) = (7, 1)$ . Each CL is embedded with 64 filter set and a stride of 3.
- The 3<sup>rd</sup> layer is the 4<sup>th</sup> CL and a primary caps layer, where the kernel size is (5, 1). Each CL is embedded with 64 filter set and a stride of 2.
- The 4<sup>th</sup> layer is the primary caps layer with 8D dimension that consists of 32 channels.
- The 5<sup>th</sup> layer is the Class Caps Layer with 16D dimension that consisting of total number of rows equalling to the total number of classes.

On the other hand, with smaller kernel size in CLs, the network performs better on short sentences. Increasing the parallel CLs to four in 2-phase CapsNet-CS results in increased accuracy than the one with three layers. This

**Table 2.** Validation accuracy results on varying kernel sizes in CLs.

CapsNet model	Kernel size		Validation accuracy
	1st CL	2nd CL	
(a) CG dataset			
1-Phase	300×(3,4,5)	3×1	83.51
	300×(3,4,5,6)	7×1	83.08
	300×(3,4,5,6)	3×1	84.02
	300×(7,8,9,10)	7×1	84.90
2-Phase	300×(7,8,9,10)	7×1	86.25
2-Phase with CS	300×(7,8,9,10)	7×1	87.16
CapsNet Model	Kernel size		Validation Accuracy
	1st CL	2nd CL	
(b) MLEE dataset			
1-Phase	300×(3,4,5)	3×1	93.26
	300×(3,4,5,6)	7×1	92.83
	300×(3,4,5,6)	3×1	93.77
	300×(7,8,9,10)	7×1	94.65
2-Phase	300×(7,8,9,10)	7×1	96
2-Phase with CS	300×(7,8,9,10)	7×1	96.91
CapsNet model	Kernel size		Validation accuracy
	1st CL	2nd CL	
(c) BioNLPST2013 dataset			
1-Phase	300×(3,4,5)	3×1	96.3
	300×(3,4,5,6)	7×1	95.87
	300×(3,4,5,6)	3×1	96.81
	300×(7,8,9,10)	7×1	97.69
2-Phase	300×(7,8,9,10)	7×1	99.04
2-Phase with CS	300×(7,8,9,10)	7×1	99.95



**Table 3.** MAPE on varying kernel sizes in CLs.

CapsNet Model	Kernel size		Validation Accuracy
	1st CL	2nd CL	
(a) CG dataset			
1-Phase	300×(3,4,5)	3×1	11.28
	300×(3,4,5,6)	7×1	11.71
	300×(3,4,5,6)	3×1	10.77
	300×(7,8,9,10)	7×1	9.89
2-Phase	300×(7,8,9,10)	7×1	8.54
2-Phase with CS	300×(7,8,9,10)	7×1	7.63
CapsNet model	Kernel size		Validation accuracy
	1st CL	2nd CL	
(b) MLEE dataset			
1-Phase	300×(3,4,5)	3×1	5.53
	300×(3,4,5,6)	7×1	5.96
	300×(3,4,5,6)	3×1	5.02
	300×(7,8,9,10)	7×1	4.14
2-Phase	300×(7,8,9,10)	7×1	2.79
2-Phase with CS	300×(7,8,9,10)	7×1	1.88
CapsNet model	Kernel size		Validation accuracy
	1st CL	2nd CL	
(c) BioNLPST2013 dataset			
1-Phase	300×(3,4,5)	3×1	4.6
	300×(3,4,5,6)	7×1	5.03
	300×(3,4,5,6)	3×1	4.09
	300×(7,8,9,10)	7×1	3.21
2-Phase	300×(7,8,9,10)	7×1	1.86
2-Phase with CS	300×(7,8,9,10)	7×1	0.95

shows increased parallelism towards the data training and data validation on all the three datasets.

#### 4.5 Error inspection

This section provides the results of sentences that are correctly or incorrectly predicted by the 2-Phase CapsNet with CS over all the three datasets. The results show that the proposed model on 2-Phase with CS achieves reduced rate of MAPE, which is lesser than 10% on all the three datasets. The result shows that the prediction accuracy of BEE is better than the previous CapsNet Models (Table 3).

300×(3,4,5) and 300×(3,4,5,6) at times provides higher rate of validation accuracy, where it is seen that with increasing channel size in CL, the accuracy increases. In other words, higher the validation accuracy, lower is the computational cost using the proposed method.

## 5. Conclusions

In this paper, the CapsNet is combined with combination strategy that helps in extraction of multi-level biomedical events. The model detects the triggers or features effectively using CapsNet that helps in classification of relations in an automated way. Such integration of deep learning with combination strategy attains effective integration of outputs at each stage, which forms the event in an optimal manner with reduced errors. The simulation conducted on various datasets shows that the CapsNet based combination strategy effectively obtains improved rate of accuracy than conventional methods. This shows the effectiveness of combining both the methods over NLP-BEE. In future, various unsupervised machine learning models can be utilised to train the network with training BEE corpora. With higher level semantics and text mining, the conversion of massive biomedical text articles can be utilised for structuring the medical related information, where the proposed model can serve as a stepping stone for the betterment of event extraction in BEE. The study shows that the 2-phase training obtains better accuracy on prediction of features than a 1-phase training of model. An additional training is essentially required by the 2-phase training model that provides higher prediction accuracy without increasing the training data. This approach hence provides a better choice for training the multi-level datasets or multi-module neural networks.

## References

- [1] Fiorini N, Canese K, Starchenko G, Kireev E, Kim W and Miller V *et al.* 2018 Best Match: new relevance search for PubMed. *PLoS Biol.* 16(8): e2005343
- [2] Raja R, Ganesan V and Dhas S G 2018 Analysis on improving the response time with PIDSARSA-RAL in CloudFlows mining platform. *EAI Endorsed Trans. Energy Web* 5(20): 1–4
- [3] Yuvaraj N and Dhas C S G 2020 High-performance link-based cluster ensemble approach for categorical data clustering. *J. Supercomput.* 76(6): 4556–4579
- [4] Daniel A, Kannan B B and Kousik N V 2021. Predicting Energy Demands Constructed on Ensemble of Classifiers. In: *Intelligent Computing and Applications* (pp. 575-583). Springer, Singapore
- [5] Sangeetha S B, Blessing N W and Sneha J A 2020. Improving the training pattern in back-propagation neural networks using holt-winters' seasonal method and gradient boosting model. In: *Applications of Machine Learning* (pp. 189-198). Springer, Singapore
- [6] Ramanan S and Nathan P S 2013. Performance and limitations of the linguistically motivated Cocoa/Peaberry system in a broad biological domain: *Citeseer*; p. 86
- [7] Li L and Jiang Y 2017. Biomedical named entity recognition based on the two channels and sentence-level reading control

- conditioned LSTM-CRF. In: *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2017
- [8] Zhu Q, Li X, Conesa A and Pereira C 2017 GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*. 34(9): 1547–54
- [9] Wang Y, Wang J, Lin H, Tang X, Zhang S and Li L 2018 Bidirectional long short-term memory with CRF for detecting biomedical event trigger in FastText semantic space. *BMC Bioinform*. 19(20): 507
- [10] Zheng S, Hao Y, Lu D, Bao H, Xu J and Hao H *et al.* 2017 Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*. 257: 59–66
- [11] Raj D, Sahu S and Anand A 2017. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics
- [12] Miwa M and Bansal M 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics; 2016
- [13] Li F, Zhang M, Fu G and Ji D 2017 A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinforma*. 18(1): 198
- [14] Kim J, Jang S, Choi S and Park E L 2018. Text classification using capsules. *arXiv 2018*, [arXiv:1808.03976](https://arxiv.org/abs/1808.03976)
- [15] Zhao W, Ye J, Yang M, Lei Z, Zhang S and Zhao Z 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv 2018*, [arXiv:1804.00538](https://arxiv.org/abs/1804.00538)
- [16] Xiao L, Zhang H, Chen W, Wang Y and Jin Y 2018. MCapsNet: Capsule Network for Text with Multi-Task Learning. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 31 October–4 November 2018
- [17] Wang Y, Sun A, Han J, Liu Y and Zhu X 2018. Sentiment Analysis by Capsules. In: *Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018*; pp. 1165–1174
- [18] Saurabh S, Prerna K and Vartika T 2018. Identifying Aggression and Toxicity in Comments using Capsule Network. In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, NM, USA, 25 August 2018; pp. 98–105
- [19] Björne J and Salakoski T 2015. TEES 2.2: biomedical event extraction for diverse corpora. *BMC Bioinforma*. 16(16):S4
- [20] Björne J, Heimonen J, Ginter F, Airola A, Pahikkala T and Salakoski T 2011 Extracting contextualized complex biological events with rich graph-based feature sets. *Comput Intell*. 27(4): 541–57
- [21] Miwa M and Ananiadou S 2015 Adaptable, high recall, event extraction system with minimal configuration. *BMC Bioinforma*. 16(10): S7
- [22] Majumder A, Ekbal A and Naskar S K 2016. Biomolecular Event Extraction using a Stacked Generalization based Classifier. In: *Proceedings of the 13th International Conference on Natural Language Processing*; p. 55–64
- [23] Li F, Ji D, Wei X and Qian T 2015. A transition-based model for jointly extracting drugs, diseases and adverse drug events. In: *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2015
- [24] Wang A, Wang J, Lin H, Zhang J, Yang Z and Xu K 2017 A multiple distributed representation method based on neural network for biomedical event extraction. *BMC Med Informa Decis Mak*. 17(3): 171
- [25] Li L, Liu Y and Qin M. Extracting Biomedical Events with Parallel Multi-Pooling Convolutional Neural Networks. *IEEE/ACM Trans Comput. Biol. Bioinforma*. 20181
- [26] Björne J and Salakoski T 2018. Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. In: *Proceedings of the BioNLP 2018 workshop*. Association for Computational Linguistics
- [27] Li L, Wan J, Zheng J and Wang J 2018 Biomedical event extraction based on GRU integrating attention mechanism. *BMC Bioinforma*. 19(9): 177
- [28] Pang B, Lee L and Vaithyanathan S 2002. Thumbs up?: Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing—Volume 10*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 79–86
- [29] Silva J, Coheur L, Mendes A C and Wichert A 2011 From symbolic to sub-symbolic information in question classification. *Artif. Intell. Rev*. 35: 137–154
- [30] Hermann K M and Blunsom P 2013. The Role of Syntax in Vector Space Models of Compositional Semantics. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 4–9 August 2013
- [31] Dufourq E and Bassett B A 2018. Automated classification of text sentiment. *arXiv 2018*, [arXiv:1804.01963](https://arxiv.org/abs/1804.01963)
- [32] Kim Y 2014. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 25–29 October 2014, pp. 1746–1751
- [33] Cheng J, Dong L and Lapata M 2016. Long short-term memory-networks for machine reading. *arXiv 2016*, [arXiv:1601.06733](https://arxiv.org/abs/1601.06733)
- [34] Kousik N, Natarajan Y, Raja R A, Kallam S, Patan R and Gandami A H 2021 Improved salient object detection using hybrid Convolution Recurrent Neural Network. *Expert Systems with Applications* 166: 114064
- [35] Cai G and Xia B 2015. Convolutional neural networks for multimedia sentiment analysis. In: *Proceedings of the 4th CCF Conference on Natural Language Processing and Chinese Computing—Volume 9362*; Springer-Verlag: Berlin/Heidelberg, Germany, 2015; pp. 159–167
- [36] Zhang X, LeCun Y. Text understanding from scratch. *arXiv 2015*, [arXiv:1502.01710](https://arxiv.org/abs/1502.01710)
- [37] Conneau A, Schwenk H, Barrault L and LeCun Y 2016. Very deep convolutional networks for natural language processing. *arXiv 2016*, [arXiv:1606.01781](https://arxiv.org/abs/1606.01781)
- [38] Lai S, Xu L, Liu K and Zhao J 2015. Recurrent convolutional neural networks for text classification. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*; AAAI Press: Austin, TX, USA, 2015; pp. 2267–2273
- [39] Sabour S, Frosst N and Hinton G E 2017. Dynamic routing between capsules. In: *Advances in Neural Information Processing Systems; Neural Information Processing Systems Foundation, Inc.*: Long Beach, CA, USA, 2017
- [40] Kim J, Jang S, Choi S, Park E L 2018. Text classification using capsules. *arXiv 2018*, [arXiv:1808.03976](https://arxiv.org/abs/1808.03976)

- [41] Xiao L, Zhang H, Chen W, Wang Y and Jin Y M 2018. CapsNet: Capsule Network for Text with Multi-Task Learning. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 31 October–4 November 2018
- [42] Raja R A, Yuvaraj N and Kousik N V 2021. Analyses on Artificial Intelligence Framework to Detect Crime Pattern. *Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications*, 119-132
- [43] Saurabh S, Prerna K and Vartika T 2018. Identifying Aggression and Toxicity in Comments using Capsule Network. In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, NM, USA, 25 August 2018; pp. 98–105
- [44] Xiang C, Zhang L, Tang Y, Zou W and Xu C 2018 MS-CapsNet: A novel multi-scale capsule network. *IEEE Signal Processing Letters* 25(12): 1850–1854
- [45] Jacob I J 2020 Performance evaluation of caps-net based multitask learning architecture for text classification. *J. Artif. Intell.* 2(01): 1–10
- [46] Xiang H, Huang Y S, Lee C H, Chien T Y C, Lee C K, Liu L and Chang R F 2021 3-D Res-CapsNet convolutional neural network on automated breast ultrasound tumor diagnosis. *Eur. J. Radiol.* 138: 109608
- [47] Toraman S, Alakus T B and Turkoglu I 2020 Convolutional capsnet: A novel artificial neural network approach to detect COVID-19 disease from X-ray images using capsule networks. *Chaos Solitons Fractals* 140: 110122
- [48] Huang W and Zhou F 2020 DA-CapsNet: dual attention mechanism capsule network. *Sci. Rep.* 10(1): 1–13
- [49] Dhiman G and Kumar V 2017 Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* 114: 48–70
- [50] Dhiman G and Kumar V 2018 Emperor penguin optimizer: A bio-inspired algorithm for engineering problem. *Knowl. Based Syst.* 159: 20–50
- [51] Kaur S, Awasthi L K, Sangal A L and Dhiman G 2020 Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* 90: 103541
- [52] Dhiman G and Kaur A 2019 STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* 82: 148–174
- [53] Kumar R and Dhiman G 2021 A comparative study of fuzzy optimization through fuzzy number. *Int. J. Mod. Res.* 1: 1–14
- [54] Chatterjee I 2021 Artificial Intelligence and Patentability: Review and Discussions. *Int. J. Mod. Res.* 1: 15–21
- [55] Vaishnav P K, Sharma S and Sharma P 2021 Analytical Review Analysis for Screening COVID-19. *Int. J. Mod. Res.* 1: 22–29
- [56] Gupta V K, Shukla S K and Rawat R S 2022 Crime tracking system and people's safety in India using machine learning approaches. *Int. J. Mod. Res.* 2(1): 1–7
- [57] Sharma T, Nair R and Gomathi S 2022 Breast Cancer Image Classification using Transfer Learning and Convolutional Neural Network. *Int. J. Mod. Res.* 2(1): 8–16
- [58] Shukla S K, Gupta V K, Joshi K, Gupta A and Singh M K 2022 Self-aware Execution Environment Model (SAE2) for the Performance Improvement of Multicore Systems. *Int. J. Mod. Res.* 2(1): 17–27