Received: 30 March 2020 | Revised: 6 June 2020 | Accepted: 6 June 2020

DOI: 10.1002/cpe.5938

WILEY

SPECIAL ISSUE PAPER

A deep learning model for mining and detecting causally related events in tweets

Humayun Kayesh[1] | Md. Saiful Islam[1] | Junhu Wang[1] | A.S.M. Kayes[2] | Paul A. Watters[2]

[1] School of Information and Communication Technology, Griffith University, Southport, Queensland, Australia

[2] School of Engineering and Mathematical Sciences, La Trobe University, Bundoora, Victoria, Australia

Correspondence
Md. Saiful Islam, School of Information and Communication Technology, Griffith University, Brisbane, QLD, Australia.
Email: saiful.islam@griffith.edu.au

Abstract

Nowadays, public gatherings and social events are an integral part of a modern city life. To run such events seamlessly, it requires real time mining and monitoring of causally related events so that the management can make informed decisions and take appropriate actions. The automatic detection of event causality from short text such as tweets could be useful for event management in this context. However, detecting event causality from tweets is a challenging task. Tweets are short, unstructured, and often written in highly informal language which lacks enough contextual information to detect causality. The existing approaches apply different techniques including hand-crafted linguistic rules and machine learning models. However, none of the approaches tackle the issue related to the lack of contextual information. In this paper, we detect event causality in tweets by applying a context word extension technique and a deep causal event detection model. The context word extension technique is driven by background knowledge extracted from one million news articles. Our model achieves 79.35% recall and 67.28% f1-score, which are 17.39% and 2.33% improvements to the state-of-the-art approach.

KEYWORDS

event causality, context word extension, feature enhancement, deep learning

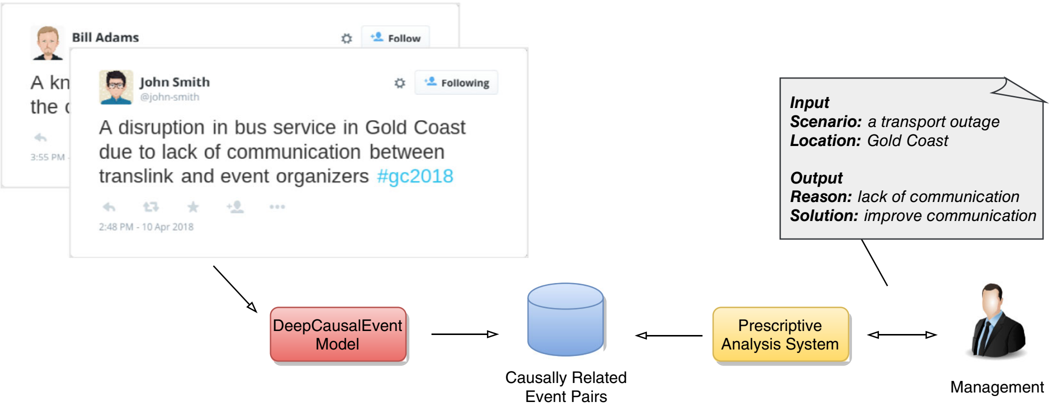
1 | INTRODUCTION

The "smart city" concept has drawn researchers' attention from various fields. The researchers have been working on improving the quality of social lives in a smart city. Public events and social gatherings are common incidents of social life in a smart city. However, often such social gatherings may be disrupted by various factors. For example, a road accident may cause transport disruptions near an event venue. Often frustrated event goers post on social media about these kinds of disruptions rather than notifying the relevant event authorities. These situations require active monitoring and timely actions from the event management authorities to run the events smoothly. Hence, a causality detection technique that can automatically discover the causes of such disruptions from social media could assist the event authorities to assess the situation and make informed decisions, for example, informing event goers to avoid certain roads and use the alternative routes. Mining social media data such as tweets could be an important source of such user-reported causally related events.

1.1 | Motivating example

Figure 1 illustrates a hypothetical example of event causality detection in tweets and its use in prescriptive analysis. The example tweet in Figure 1: "A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers" contains two causally related events.

© 2020 John Wiley & Sons, Ltd.



A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers #gc2018

Input
Scenario: *a transport outage*
Location: *Gold Coast*

Output
Reason: *lack of communication*
Solution: *improve communication*

FIGURE 1 Application of automatic event causality detection in predictive event analysis (adapted from Kayesh et al[7])

Here, the "lack of communication" is a causal event and "a disruption in bus service in Gold Coast" is the corresponding effect event. A prescriptive analysis system, build on a dataset of such causally related event pairs, can help the authorities to plan the public transport services offered to the city dwellers better and minimize the chance of transport outage in the future.

1.2 | Applications

Other application areas of causality detection in social media include political events analysis[1], income analysis[2], career path prediction[3], adverse drug reaction (ADR) detection[4], and automatic question answering[5]. For instance, postmarketing surveillance of drugs is a vital activity of the drug safety authority. The surveillance is often dependent on the ADR-related responses from the doctors and patients. However, not many doctors have time to report all the ADR cases they observe. Few patients feel motivated to fill up a long form for self-reporting ADR experiences. Hence, all the ADRs for a drug might not be identified timely. This challenge can be tackled by automatically detecting drug names (causes) and ADRs (effects) in social media posts because a large number of social media posts contains medical information[6].

1.3 | Challenges

Event causality detection is a challenging natural language processing task and it is an evolving area of research[8-15]. In fact, causality detection in short text such as tweets is more challenging than relatively more formal text such as news articles[16]. The existing approaches that apply hand-crafted rules[14,17,18] are less effective in tweets (see Section 5) compared to news articles. The linguistic rules-based approaches expect text to be grammatically correct but tweets are highly informal and prone to grammatical errors. Another challenge of event causality detection in tweets is lack of contextual information. Tweets are short and often contextual information is missing, which makes it difficult to develop machine learning-based models. For example, a machine learning model proposed by Ponti et al[19] is not much effective on detecting event causality in tweets. To deal with this issue, in this paper we propose to apply a context word extension technique that can add additional contextual words based on background knowledge.

1.4 | Contributions

In this paper, we propose an improved version of our preliminary work[7] (technical report[20]) on event context word extension technique and neural network model for mining causally related social events in tweets. In our earlier work, the representation of causal and effect events does not maintain the original order of the event context words and thereafter, results in information loss. To tackle this issue, we propose a new sequence-aware representation of causal and effect events which retains the original order of words as in the original causal and effect phrases in tweets. Additionally, the previous approach merges the causal and effect event words together which simplifies the feature space and reduces the separability of causally related event pairs from other pairs. In the new approach, we extract separate features for causal and effect events by applying bidirectional long short-term memory (BLSTM) and multihead attention mechanism. Finally, we apply two parallel two-dimensional (2D) convolutional neural networks (CNNs) followed by 2D max-pooling layers to extract features from the combined casual features and effect features. A softmax layer is then used to produce the final output. The new approach outperforms our previous approach and the state-of-the-art. The main contributions of this paper are outlined below:

1. we propose a sequence-aware event representation technique to represent candidate causal and effect events;
2. we propose a novel technique to extend event context words by applying background knowledge to detect causality in tweets;
3. we develop a deep neural network model that extracts causal features separately from candidate causal and effect events; and
4. we perform extensive experiments to compare our model with the existing state-of-the-art models for causality detection in short text.

1.5 | Organization

We present the remaining sections of this paper in the following order: Section 2 describes the related work; Section 3 discusses the formal definition of the research problem investigated in this work; the proposed approach to event causality detection in tweets is described in Section 4; the experimental results and discussions are illustrated in Section 5, and the conclusion remarks are presented in Section 6.

2 | RELATED WORK

In the literature, we find different approaches to detect causality in texts. These approaches can be grouped into two broad categories: (i) phrase-based causality detection and (ii) event-based causality detection. We include all the approaches that detect causality from candidate causal phrases in the first group and the approaches that extract candidate causal events before detecting causality in the second group.

2.1 | Phrase-based causality detection

The phrase-based causality detection approaches mostly focus on word-to-word relation and co-occurrence of words as the features to detect causality. Riaz et al[11] propose a technique to detect causality by extracting relationship between verb-verb pairs. In this work, the authors train a supervised machine learning model on an automatically extracted dataset of verb-verb causal relationships. An improved version of this work[12] proposed by the same authors utilizes noun-verb relationship in a sentence. This technique automatically extracts grammatically connected noun-verb pairs and then, like the previous approach, applies a machine learning model to detect causal relationship between the pairs. The machine learning model is trained on both lexical and semantic features. The authors also apply the structural features of sentences to train the model. Another phrase-based approach is proposed by Luo et al[14] that applies linguistic rules and commonsense knowledge extracted from web text to detect causality between two candidate causal phrases. The commonsense knowledge is stored in the form of a causal network. The causal network contains the frequencies of word co-occurrence in a causal sentence extracted from the web text. These scores are used to calculate the causal strength between two candidate cause and effect phrases. The causal network preparation in this work is improved by a technique proposed by Sasaki et al[17] that includes multiword expressions in the causal network. Recently, Yu et al[21] propose a neural network based approach to detect causality in science publications articles. The authors apply Bidirectional Encoder Representations from Transformers (BERT) model on a manually labeled dataset of 3000 conclusion sentences extracted from PubMed articles. Another recent approach proposed by Doan et al[18] applies hand-crafted linguistic rules to extract health-related causality from twitter data. The approaches mentioned above aim to detect causality between pairs of causal phrases but these approaches do not consider events before detecting casual relationship.

2.2 | Event-based causality detection

Some existing causality detection approaches extract events related information from text and detect causality between event pairs[9,10,13,15,22-25]. Do et al[9] proposes a probabilistic approach to event causality detection that applies point-wise mutual information (PMI) score to calculate the causal strength between two events. At first, the authors extract candidate causal event and effect event. Each event contains an event keyword and a set of event attributes. Then, the authors calculate PMI scores between keyword-keyword, keyword-attribute, and attribute-attribute pairs between candidate causal and effect events. The authors apply inverse document frequency score, discourse relations and sentence structural information such as word-to-word distance in the sentence as the additional features in their event causality detection model. In this approach, *Penn discourse treebank*[22] is used to detect discourse relation. These association mining techniques work well for frequent events but causal events in tweets are not so frequent, hence such techniques are not suitable for tweets[24].

Mirza[13] proposes another event causality detection approach that extracts both causal and temporal relations between event pairs. The author also proposes a set of rules to annotate event pairs to prepare an event causality dataset. The approach assumes that a causal event should appear before the effect event in the text. However, the temporal information is often missing in tweets. Additionally, this approach accepts datasets that are annotated using the guideline proposed in this work which makes the proposed approach less applicable to other datasets.

Some recent approaches apply neural networks to detect event causality[15,19]. Kruengkrai et al[15] proposes an event causality detection approach that applies background knowledge and multicolumn neural network[10]. The authors use word vectors as features to train the neural network model. Rahimtoroghi et al[25] extract causally related event pairs from user-generated text by applying co-occurrence between events. The authors calculate causal potential between a pair of candidate causal and effect event by calculating the probabilistic adjacency score. To determine adjacency score between two events, this technique applies 2-skip bi-gram model. A similar approach is proposed by Khan et al[26]. At first, this approach detects causally related event pairs in a time series dataset of system event logs by applying association rule mining technique. Then it prepares a chain of causally related evens by merging two event pairs if they have the same effect event. However, this approach ignores any causal relationship between the causal events in the event pairs when merging two event pairs into a causal chain.

Though there are a number of other recent event causality detection approaches[4,27-29], Ponti et al[19] proposes a feature enhancement technique and a neural network-based model that is most relevant to our work. The authors apply word-to-word distance to enhance the feature set while training the neural network model. The authors assume that the distance between event keyword and attribute words contains useful information to detect causality. However, these positional features are not much effective for event causality detection in highly informal text such as tweets (please see Section 5).

3 | PROBLEM FORMULATION

We define an event as representation of an incident and an event consists of two types of words: an event keyword and a set of event attribute words. An event keyword is the word that can contain majority of the information to represent the event. The attribute words are the other words that are grammatically related to the event keyword. Table 1 displays the event representation format with some example events. When an event e_1 (directly or indirectly) causes another event e_2 to happen, we denote e_1 to be the causal event and e_2 to be the effect event. For instance, in the following sentence: "A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers," *lack of communication* represents e_1, and *disruption in bus service* represents e_2. In this paper, we aim to automatically detect this kind of causally related event pairs in tweets as shown in Figure 1. To be specific, we investigate the following research question:

RQ: *How can we automatically detect pairs of cause and effect events in tweets?*

Informal and unstructured nature of tweets is the main challenge of this task. Hence, simple linguistic rule-based approaches[14,17] see poor performance in twitter datasets. Also, contextual information is often missing in short text such as tweets. For this reason machine learning-based approaches, for example the approach proposed by Ponti et al[19], often fail to detect causal relationship between events in tweets. Hence, we propose an event context word extension technique to automatically detect causal relationship between candidate causal event e_1 and effect event e_2. Formally, the problem investigated in this paper can be defined as follows:

$$f(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 \text{ causes } e_2, \\ 0, & \text{otherwise} \end{cases}$$

where "1" represents the "Causal" relationship and "0" represents the "Not Causal" relationship between events in a candidate cause-effect event pair. A summary of notations and symbols used in this paper are given in Table 2.

TABLE 1 Representation of events (adapted from Kayesh et al[7])

Sentences	Events
Storm **hits** Gold Coast	hit (storm, gold, coast)
Mike **crashed** his car in Gold Coast	crash (mike, car, gold, coast)
Heavy traffic **jam** in Gold Coast today	jam (traffic, coast, gold, today)
A **disruption** in bus service in Gold Coast due to **lack** of communication translink and event organizers	disruption (bus, service, coast, gold) lack (communication, organizer, translink)

TABLE 2 Summary of notations and symbols

Notations	Descriptions
e_1	Causal event
e_2	Effect event
e'_1	Context word extended causal event
e'_2	Context word extended effect event
w_k^c	Causal event keyword
w_p^c	Causal event context word
w_k^e	Effect event keyword
w_q^e	Effect event context word
w_{ex}^c	A list of extended causal context words
w_{ex}^e	A list of extended effect context words
$w_{ex_n}^c$	An extended causal context word
$w_{ex_n}^e$	An extended effect context word
v_k^c	Embedding vector of the causal keyword
v_p^c	Embedding vector of a causal context word
v_k^e	Embedding vector of the effect keyword
v_q^e	Embedding vector of an effect context word
$v_{ex_n}^c$	Embedding vector of an extended causal context word
$v_{ex_n}^e$	Embedding vector of an extended effect context word
$\rightarrow$	Causes

4 | OUR APPROACH

In this paper, we propose a deep causal event detection model for detecting event causality in tweets using the event context word extension technique. At first, we perform a number of necessary preprocessing operations on tweets and identify the pairs of candidate cause and effect events. Unlike our previous work[7,20], we then represent event phrases as events using a sequence aware event representation technique that retains the original order of words in a tweet. We believe that background knowledge is essential in causality detection, and hence we extract background knowledge from news articles and build a causal background knowledge network of causally related words. We utilize this network to extend event context words. We then extract causal and effect features separately from extended causal and effect events. We then combine the causal and effect features and apply two set of CNNs and max pooling operations parallelly on them. The outputs are then combined and flattened to produce a single vector. This vector is then passed to a dropout layer followed by a softmax layer to produce the final label. Figure 2 displays a high-level overview of the workflow in the proposed approach.

4.1 | Tweet preprocessing

Tweet preprocessing is the first step in out approach. This preprocessing step is aimed to reduce noisy characters without sacrificing any useful information. In this work we assume that a cause-effect event pair appears in a single sentence, hence we consider tweets as the bag of sentences. At first, we split tweets into individual sentences and discard emojis, hashtags (#), and "@" characters. If a word contains repeating characters, we change that word into a normalized version, for example, "yesss" is converted to "yes." We also remove URLs and the sentences with question marks. After preprocessing the tweets, we pass the sentences to the candidate event pair extraction and representation stage.

4.2 | Sequence-aware event pair extraction and representation

In this stage we extract candidate causal and effect event pairs from a sentence. We propose a sequence aware event representation technique that retains the original order of words in the event representations. As a first step to extract events, we split a sentence into candidate causal and effect

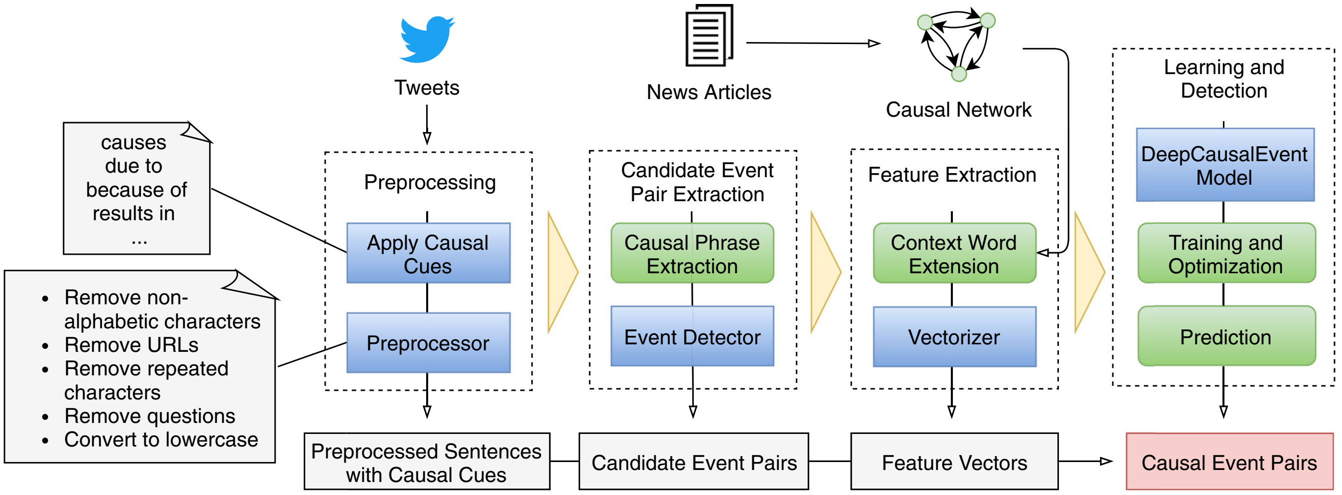


FIGURE 2 An overview of our sequence-aware deep causal event detection model (adapted from Kayesh et al[7])

TABLE 3 Causal cue words used for candidate causal and effect phrases extraction from tweets[7]

Affect	Brought On	Due To	Increased By	Reason Of
affected by	cause	effect of	increases	reasons of
affects	caused	for this reason alone	induce	result from
and consequently	caused by	gave rise to	induced	resulted from
and hence	causes	give rise to	inducing	resulting from
as a consequence	causing	given rise to	lead(s) to	results from
as a consequence of	consequently	giving rise to	leading to	so that
as a result of	coz	hence	led to	that's why
because	coz of	in consequence of	on account of	the result is
because of	decrease	in response to	owing to	thereby
bring on	decreased by	inasmuch as	reason for	therefor
brings on	decreases	increase	reasons for	thus

phrases. Table 3 shows the causal cue words that we use to extract the phrases. For instance, the sentence: "a disruption in bus service in gold coast due to lack of communication between translink and event organizers" is split into two phrases where "lack of communication between translink and event organizers" is the candidate causal phrase and "a disruption in bus service in gold coast" is the candidate effect phrase. Here, "due to" is the cue word used to split the sentence. We then represent both of the candidate phrases by a sequence of contextual words using the sequence aware event representation technique. The candidate causal event is represented as $w_k^c(w_0^c, w_1^c, w_2^c \ldots w_p^c)$ where w_k^c is the event keyword and w_p^c is an event context word. Similarly, the candidate effect event is represented as $w_k^e(w_0^e, w_1^e, w_2^e \ldots w_q^e)$ where w_k^e is the event keyword and w_q^e is an event context word. An event key word is considered to be the trigger word of the event and event context words are the other words grammatically related to the event keyword. To extract the event keyword and the context words from a candidate event phrase we apply Stanford dependency parser[30]. As proposed in Kayesh et al[7], the root word in the dependency relations is considered to be the event keyword and the other words linked to the root word via any of the following relations: "nsubj," "nsubjpass," "amod," "dobj," "advmod," "nmod," "xcomp," "compound:prt," "compound," and "neg," is considered to be a context word. Unlike our previous works[7,20], we preserve the original order of the context words in the event representation. For example, "A disruption in bus service in Gold Coast" is represented as "disruption(bus, service, gold, coast)" where *disruption* is the event keyword and *bus*, *service*, *gold*, and *coast* are the event context words. The event context words appear in the same order as they are shown in the event representation. Figure 3 shows the process of extracting event pairs from the sentence "A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers."

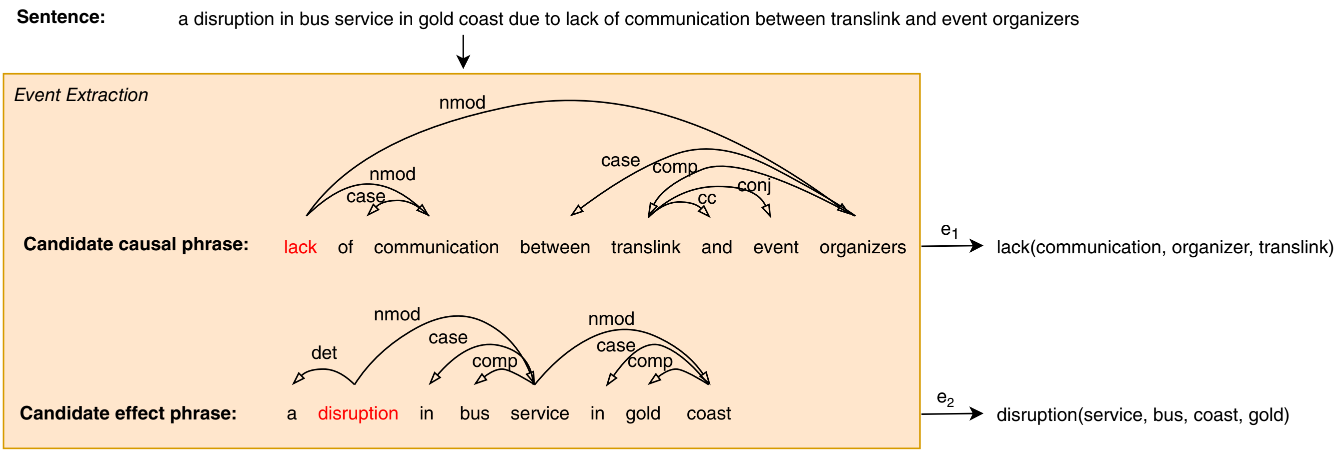

Sentence:
a disruption in bus service in gold coast due to lack of communication between translink and event organizers
Event Extraction
nmod
nmod
case
case
comp
cc
conj
Candidate causal phrase:
lack of communication between translink and event organizers
e1
lack(communication, organizer, translink)
det
nmod
case
comp
nmod
case
comp
Candidate effect phrase:
a disruption in bus service in gold coast
e2
disruption(service, bus, coast, gold)

FIGURE 3 An example of event pair extraction from a sentence[7]

TABLE 4 Linguistic rules used for causal background knowledge network generation

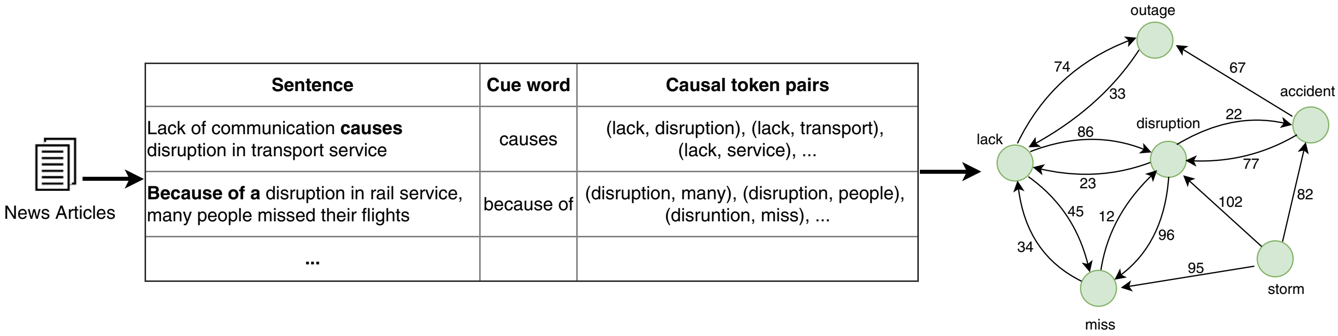
B cue_words A	A cue_words B		cue_words A, cue_words B	cue_words B, cue_words A
, because	**and consequently**	**given rise to**	**if … ,**	**the reason for … , was**
as a consequence of	and hence	giving rise to	if … , then	the reason of … , is
as a result of	bring on	hence	in consequence of … ,	the reason of … , was
because	bringing on	induce	owing to … ,	the reasons for … , are
because of	brings on	induced	the effect of … , is	the reasons for … , were
caused by	brought on	induces	the effect of … , was	the reasons of … , are
due to	cause	inducing	the effect of … , will	
in consequence of	caused	lead to		
inasmuch as	causes	leading to		
owing to	causing	leads to		
result from	consequently	led to		
resulting from	for this reason alone	therefore		
results from	gave rise to	thus		
results from	give rise to			

4.3 | Causal network

News paper articles are a good source of causality-related background knowledge[18,21,27-29,31-33]. Hence, we capture background knowledge on causality and build a causal background knowledge network from a set of one million news articles[a] by following the technique proposed by Luo et al[14]. The dataset is prepared by collecting news articles between the period of November 1, 2015 to November 30, 2015. More than 95K unique source websites are used to collect articles. On an average, an article in the dataset contains 405 words[34].

To build the causal background knowledge network, we split the news articles into sentences and extract causal and effect phrases by applying the linguistic rules displayed in Table 4. For example, if there is a sentence "Because of a disruption in rail service, many people missed their flights" in a news article, we extract "a disruption in rail service" as the causal phrase and "many people missed their flights" as the effect phrase. Newspaper articles are considered to be more formal and grammatically more correct then tweets. Since we use the causal sentences from newspaper articles to train our model, we apply a refined and less ambiguous list of cue words (compared to Table 3) to extract causal and effect phrases from news paper articles. We tokenize and lemmatize the causal and effect phrases and then use each word as a vertex in a directed graph which we refer as the causal background knowledge network. The edges of the network contain the frequency of a causal relationship between two vertices. For example,

[a] https://research.signalmedia.co/newsir16/signal-dataset.html



Sentence	Cue word	Causal token pairs
Lack of communication **causes** disruption in transport service	causes	(lack, disruption), (lack, transport), (lack, service), ...
Because of a disruption in rail service, many people missed their flights	because of	(disruption, many), (disruption, people), (disruntion, miss), ...
...		

FIGURE 4 Causal network construction from news articles[7]

if there is an edge from vertex A to vertex B with a value 35, it represents that there has been 35 cases when the word A and word B appeared in the causal phrase and effect phrase, respectively. The process of building causal background knowledge network from news articles is shown in Figure 4.

4.4 | Context word extension

In this step, we discuss our technique to utilize background knowledge encoded in the causal network built in Section 4.3. We apply a context word extension technique[7] that extends event context words by adding relevant words from background. The context word extension technique adds new but relevant contextual words in the event, which helps to solve the lack of context words problem in events. For example, if w_k^c is the event keyword of candidate causal event e_1 and w_k^c is the event keywords of a candidate effect events e_2, then we extend the event context words of e_1 and e_2 by using w_k^c and w_k^e and the causal network. To extend the context words $(w_0^c, w_1^c, w_2^c \dots w_p^c)$ of candidate causal event e_1, we extract a list of top n causes of w_k^e from the causal network. Similarly, we extend the context words $(w_0^e, w_1^e, w_2^e \dots w_q^e)$ of e_2 by extracting the same number of effects of w_k^c from the network. A n-word extended candidate causal event $e_1 = \{w_k^c, (w_0^c, w_1^c, w_2^c \dots w_p^c), (w_{ex_0}^c, w_{ex_1}^c, \dots w_{ex_n}^c)\}$ and candidate effect event $e_2 = \{w_k^e, (w_0^e, w_1^e, w_2^e \dots w_q^e), (w_{ex_0}^e, w_{ex_1}^e, \dots w_{ex_n}^e)\}$ where $w_{ex_n}^c$ is an extended causal context word and $w_{ex_n}^e$ is an extended effect context word. The above approach of event context word is pseudocoded in Algorithm 1 and an example of context word extension where $n = 2$ is shown in Figure 5.

Algorithm 1. Context Word Extension

```
function CONTEXT_WORD_EXTENSION ($e_1$: candidate causal event, $e_2$: candidate effect event, $n$: number of context word extension, $cnet$: causal network)
    $w_k^c \leftarrow get\_event\_keyword(e_1)$
    $w_k^e \leftarrow get\_event\_keyword(e_2)$
    $ct \leftarrow get\_causal\_terms(cnet, w_k^c)$        ▷ Returns a list of terms sorted in descending order of frequencies
    $et \leftarrow get\_effect\_terms(cnet, w_k^e)$        ▷ Returns a list of terms sorted in descending order of frequencies
    $w_{ex}^c \leftarrow list()$
    $w_{ex}^e \leftarrow list()$
    for $i \leftarrow 0$ to $n - 1$ do
        $w_{ex}^c \leftarrow w_{ex}^c + list(ct[i])$        ▷ Append terms to list $w_{ex}^c$
        $w_{ex}^e \leftarrow w_{ex}^e + list(et[i])$        ▷ Append terms to list $w_{ex}^e$
    end for
    $e_1' \leftarrow list((w_k^c), get\_context\_words(e_1), w_{ex}^c)$        ▷ Create a list with event keyword $w_k^c$ and event context words
    $e_2' \leftarrow list((w_k^e), get\_context\_words(e_2), w_{ex}^e)$        ▷ Create a list with event keyword $w_k^e$ and event context words
    return $(e_1', e_2')$
end function
```

4.5 | Causal event detection

In this section we discuss the proposed deep causal event detection model that we use in social event mining. The model has two major modules: vectorization module and the deep causal detection module. We convert the candidate causal and effect events into vector in the vectorization

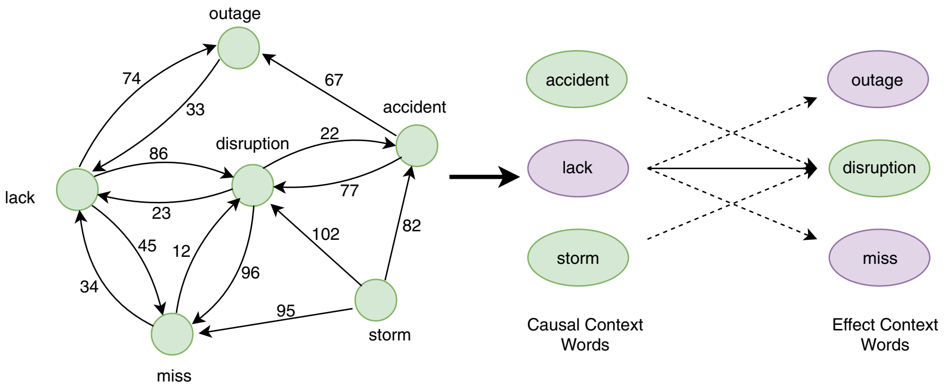


FIGURE 5 An example of *n*-word context word extension, where $n = 2$ and the original candidate cause and effect keywords are *lack* and *disruption*, respectively[7]

module. The deep causal event module extracts causal features from extended candidate causal and effect events and performs training on the training dataset to detect event causality in tweets.

4.5.1 | Vectorization

In this step, we vectorize our candidate events e_1 and e_2 into embedding vectors. We use a pretrained Word2vec model[35] to convert each word in e_1 and e_2 into a 300-dimension dense vector. At first, we extract a word-to-index dictionary D and an index-to-embedding dictionary M. The word-to-index dictionary D contains a set of key-value pairs where every pairs has a word as the key and an index number as the value. The index-to-embedding dictionary M contains pairs of a word index it is corresponding 300-dimension embedding vector. We use dictionary D to get the corresponding index of each word in e_1 and e_2. We also use padding for both context words and extended context words sequences. We then use the index-to-embedding dictionary M to convert each index in e_1 and e_2 by its respective embedding vectors. We refer to the embedding vectors in e_1 as $\{v_k^c, (v_0^c, v_1^c, v_2^c \dots v_p^c), (v_{ex_0}^c, v_{ex_1}^c, \dots v_{ex_n}^c)\}$ and e_2 as $\{v_k^e, (v_0^e, v_1^e, v_2^e \dots v_q^e), (v_{ex_0}^e, v_{ex_1}^e, \dots v_{ex_n}^e)\}$. After the conversion of causal and effect events into vectors, they are sent to the deep causal event detection model.

4.5.2 | The proposed deep causal event detection model

Our deep causal event detection model has two stages. In the first stage, we extract features from $e_1 = \{v_k^c, (v_0^c, v_1^c, v_2^c \dots v_p^c), (v_{ex_0}^c, v_{ex_1}^c, \dots v_{ex_n}^c)\}$ and $e_2 = \{v_k^e, (v_0^e, v_1^e, v_2^e \dots v_q^e), (v_{ex_0}^e, v_{ex_1}^e, \dots v_{ex_n}^e)\}$ separately and then combine the extracted features together. We extract the causal features by applying a BLSTM followed by a multihead attention model on $(v_0^c, v_1^c, v_2^c \dots v_p^c)$. Then we apply dense layers separately on v_k^c and $(v_{ex_0}^c, v_{ex_1}^c, \dots v_{ex_n}^c)$. We concatenate features for event keyword, event context words and the extended words to prepare the causal features. Similarly, we extract effect features by applying a BLSTM followed by a multihead attention model on $(v_0^e, v_1^e, v_2^e \dots v_q^e)$ and then applying dense layers on v_k^e, and $(v_{ex_0}^e, v_{ex_1}^e, \dots v_{ex_n}^e)$. The causal features and effect features are then combined together and passed to the second stage to the model.

In the second stage of the proposed deep causal event detection model, we apply two parallel 2D CNN models on the combined features to detect causality. Each CNN models are followed by a 2D max pooling layer. The output of the max pooling layers are then concatenated and flattened to generate a single vector feature. To prevent our model from over-fitting we add a dropout layer on the single vector features. The output of the dropout layer is then sent to a softmax layer to generate the final label. The label denotes whether a candidate event pairs are causal. The above model is illustrated in Figure 6.

5 | EXPERIMENTS

We perform our experiments on a manually annotated twitter dataset. Our dataset preparation process, experiment setup and outcomes are discussed in detail in this section.

5.1 | Dataset

We prepare our dataset by collecting more than 207k tweets using twitter API[b]. We only collect the tweets the are published within the date range between October 5, 2017 and May 7, 2018. We aim to collect the tweets that were related to the Commonwealth Games 2018 held in

[b] https://developer.twitter.com/en/docs/tweets/search/overview

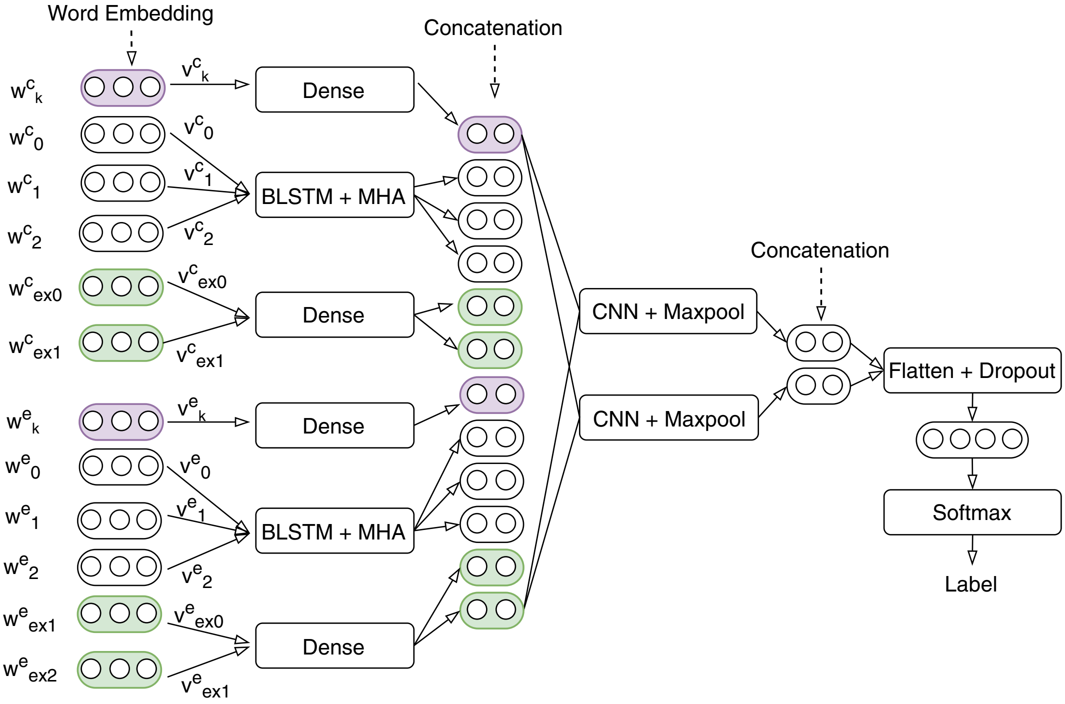


FIGURE 6 Architecture of the proposed deep causal event detection model

TABLE 5 Statistics of the tested dataset [7]

Set	Causal	Not Causal
Full dataset	459	457
Training	275	274
Test	184	183

Australia. Hence, we use a set of relevant hashtags as the keywords to collect tweets. Our set of hashtags consists of “#CommonwealthGames,” “#CommonwealthGames2018,” “#GC2018,” and “#ShareTheDream.” After collecting tweets we perform some necessary preprocessing (please see Section 4.1) and then we extract 913 pairs of candidate causal and effect events pairs by following the technique described in Section 4.2. We then manually label each pair as “Causal” if there is a causal relationship between the events, and not “Not Causal” otherwise. After annotation, we separate our training and test dataset for the experiment. We train and optimize our model parameters on 60% data, and we use the remaining 40% data for testing. While splitting our dataset randomly into training and test sets we apply stratification so that the percentage of causal and not causal event pairs remains same in both train and test sets. We illustrate the dataset statistics in Table 5.

5.2 | Setup

We implement the proposed deep causal event detection model in python 3.6 language using the Keras[c] python package. We run all experiments on a Linux 18.04 Core i7 4.2GHz PC with 32GB RAM. We use padding if any candidate causal event has less context words than the maximum length of context words in all the candidate causal events. Similarly, we pad the effect context words sequences using the maximum length of the event context words in all the candidate effect words. The dense layers used for event keywords and extended context words apply the “ReLU” activation function. In the BLSTM model for candidate causal event context words, we use activation function “tanh” and the number of units is set to 100. We set both dropout and recurrent dropout to 0.1. This BLSTM model is followed by a 100-unit multihead attention model. We use the same configuration for the BLSTM model and the multihead attention model used for the effect event context words. In the parallel 2D CNN stage, we set the number of filters to 100, kernel size to 3, and use activation function “tanh” for the the first CNN model. The 2D max pooling layer that follows this CNN model has a pool size as shown in Eq. [2], where *S* is the maximum sequence length, *Kr* is the kernel size used in the CNN model. For the other 2D CNN model,

[c] https://keras.io/

TABLE 6 Summary of model parameters

Parameter	Value
Number of filters in parallel 2D CNNs	100
Kernel size of parallel 2D CNNs	3 and 4
Dense layer activation function	ReLU
Multihead attention units	100
BLSTM units	100
BLSTM activation function	tanh
Dropout	0.10
Optimizer	adam
Loss function	Binary crossentropy
Validation metric	Accuracy
Batch size	32
Number of epochs	4

Abbreviations: 2D, two dimensional; BLSTM, bidirectional long short-term memory; CNN, convolutional neural networks.

we set the number of filters to 100, kernel size to 4 and use activation function "tanh". Similar to the previous max pooling layer we use Equation (2) to determine the pool size.

$$\text{poolsize} = (S - Kr + 1, 1). \tag{2}$$

The dropout layer before "Softmax" layer uses 10% as the dropout rate. We optimize our model by using Adam optimizer, "binary crossentropy" as the loss function, and "accuracy" as the validation metric. We train the model for four epochs with batch size set to 32. We have finalised these parameters empirically. A summary of the model parameters are given in Table 6.

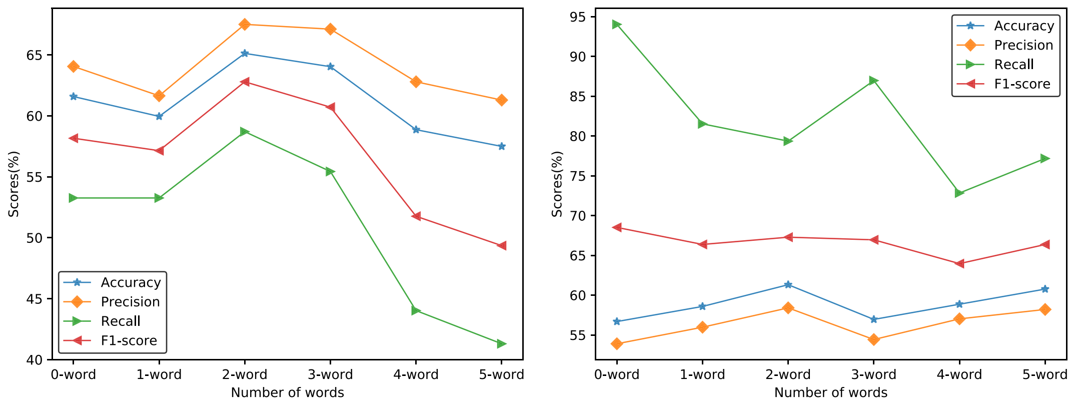
5.3 | Performance evaluation

We compare the performance of your model for different number of context word extension while keeping our deep neural network model settings same. Table 7 and Figure 7 show the results of the experiment. From our experiment we find that the 2-word extension model achieves the best accuracy 61.31% and precision 58.04% scores compared to the other context word extensions. Also, comparing against the 0-word or no-word extension model, the 2-word extension model achieves 4.63% higher accuracy and 4.51% higher recall. The high recall (94.02%) and a comparatively lower precision (53.89%) of 0-word extension model suggests that the models tends to label most of candidate causal events to causal regardless of causal relationship between events. The rationale behind choosing the 2-word extension is also evident from the Figure 8, which illustrates the receiver operating characteristic curve (ROC) curve values for the same experiment settings.

To demonstrate the effectiveness of the proposed model, we compare the proposed model DeepCausalEvent with a number of existing state-of-the-art approaches. We compare our approach with Luo et al's[14] commonsense-based approach (Commonsense), which applies commonsense knowledge to detect causality. This approach builds a causal network of commonsense and calculates causal strength between two phrases

TABLE 7 Comparison between *n*-word extensions

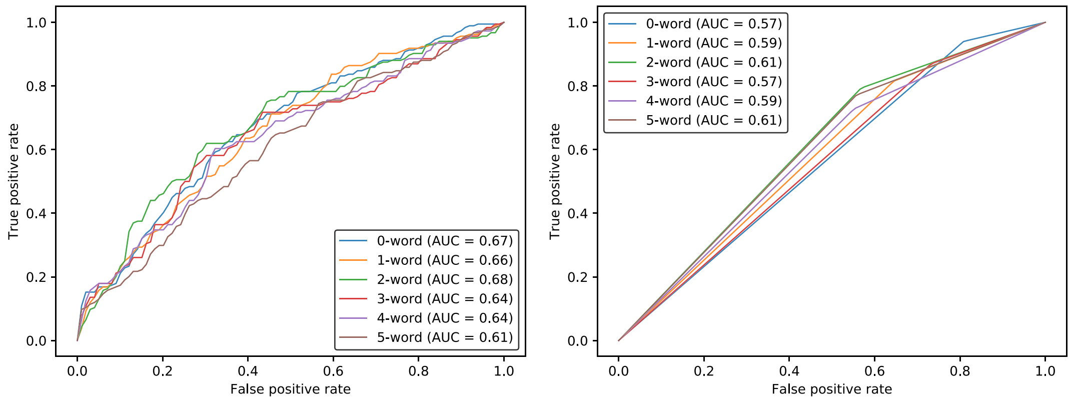
Extension	Accuracy	Precision	Recall	f1-score
0-word	56.68	53.89	94.02	68.51
1-word	58.58	55.97	81.52	66.37
2-word	61.31	58.40	79.35	67.28
3-word	56.95	54.42	86.96	66.95
4-word	58.86	57.02	72.83	63.96
5-word	60.76	58.20	77.17	66.36



(A) FFNN + n-word ext.

(B) DeepCausalEvent

FIGURE 7 The effect of number of extended event context words on the performance of the event causality detection models: (A) FFNN+*n*-word Ext. model (adapted from Kayesh et al[7]) and (B) DeepCausalEvent model



(A) FFNN + n-word ext.

(B) DeepCausalEvent

FIGURE 8 The comparison of area under the curve (AUC) values among different number of context word extensions: (A) FFNN+*n*-word ext. (adapted from Kayesh et al[7]) and (B) DeepCausalEvent

using that causal network. We also compare our approach with Sasaki et al's[17] "Commonsense + Multi-word," that extends Luo et al's approach by proposing a multiword casual network instead of single-word network. Another benchmark approach (FFNN + Position) is a neural network appraoch proposed by Ponti et al[19]. This approach proposes a feature enhancement technique that applies the word-positional features to train a feed forward neural network (FFNN) model. We train the "FFNN + Position" model for 150 epochs with learning rate 0.1 and batch size 1. We also compare the proposed model with our previous work[7] (FFNN + 2-word Ext.), which trains a FFNN model using an event context word extension technique to perform event causality detection in tweets. We implement the 2-word extension model of "FFNN + 2-word Ext." as this model is reported to be the best performer in the experiment. We also implement a variant of the DeepCasualEvent that uses pretrained BERT embedding instead of Word2vec. We refer to this model as "DeepCausalEvent+BERT Embd." When comparing against the benchmark approaches we observe that our proposed model DeepCausalEvent outperforms all the benchmark models in terms of recall and f1-score. Our proposed model achieves 79.35% recall and 67.28% f1-score. The best performer among the benchmark approaches in our experiment is *FFNN+2-word Ext*. Our proposed model achieves at least 17.39% improvement in recall and 2.33% improvement in f1-score compare to this approach. This result shows superiority of our approach over the existing state-of-the-art approaches.

The average preprocessing time of the proposed DeepCausalEvent model is 0.15 seconds for a candidate cause-effect pair while the average prediction time is 0.0016 seconds and the training time is 10.19 seconds. In real time, our system will take approximately 0.1516 seconds to detect causality and causally related events in a tweet.

The improvement of event causality detection model by applying a context word extension technique and deep neural networks is one of the key findings of this work. Table 8 shows the comparison of results among the proposed DeepCausalEvent model and other state-of-the-arts. The table shows that the commonsense-based techniques, Commonsense[14] and Commonsense + Multi-word[17], achieve low recall hence their f1-scores are low compared to the neural network-based approaches. These approaches rely on word co-occurrence and commonsense which often do

TABLE 8 Comparison of the proposed DeepCausalEvent method with existing approaches

Methods	Accuracy	Precision	Recall	f1-score
Commonsense[14]	50.95	56.67	9.24	15.89
Commonsense + Multi-word[17]	50.14	54.55	3.26	6.15
FFNN + Position[19]	59.40	60.12	56.52	58.26
FFNN + 2-word Ext.[7]	**65.94**	**67.46**	61.96	64.59
DeepCausalEvent + BERT Embd.	57.22	55.65	72.28	62.88
DeepCausalEvent	61.31	58.40	**79.35**	**67.28**

Abbreviations: FFNN, feed forward neural network.

TABLE 9 Some examples of candidate causal pairs (causal → effect) and their predicted labels by different methods including our DeepCausalEvent method—the labels “1” and “0” represent “Causal” and “Not Causal” relations, respectively and the column “Gold Data” shows the ground truth data

Candidate Causal Pairs	Gold	Commonsense	Commonsense + Multiword	FFNN + Position	FFNN + 2-word-Ex	Deep Causal Event
Persistent achiles injury→disapointed @salypearson won't be runing at #gc2018 #comonwealthgames	1	0	0	1	1	1
Samoa's don opeloge lifts 191kg→he wins	1	0	0	0	1	1
He does not know beyond cricket→homework neded	1	0	0	0	1	1
No tickets→babita's father mised her #comonwealth games2018-match	1	0	0	1	1	1
You can't do much wrong → even if you try	0	0	0	0	0	0
Imoral atack on syrian childrens→@cni trump should be impeached and hanged til death	0	0	0	1	**1**	**0**
Her father's name is not cleared as εan oficialε→she wil not take part in #gc2018	1	0	0	1	1	1
Any australian boxer is fated to win comonwealth games gold this wek→it is skye nicolson	0	0	0	0	0	0
Presure from the defence → a lose pas from malawi	1	0	0	1	1	1
He had at least asked→as he was runing	0	1	1	0	0	0
#ipl has entertainment value→india at #gc2018 fils us with pride	0	0	0	1	0	0
I want to watch it al live → is there any legislation i can use to work from home until the #comonwealthgames2018 finishes	0	1	0	1	0	0
You're in the area → please be aware there wil also be road closures and parking restrictions on competition days on 8	0	1	0	0	1	1
A technical isue → 34am central to varsity lakes train is delayed 30 minutes	1	1	1	1	1	1
The task of carying your country's flag embodies the values and ideals it represents → we have found a perfect role model for its cause	1	1	0	1	**0**	**1**
#cameronvanderburgh → big upset at #comonwealthgames	1	0	0	0	0	0

Abbreviations: FFNN, feed forward neural network.

represent causality in real-life events. For instance, “her father's name is not cleared as an official → she will not take part in #gc2018” is a causal event and the commonsense-based approaches often fail to detect these kind of event causality relationships. Though our previous model FFNN + 2-word ext.[7] achieves better accuracy and precision scores (results are shown in bold in Table 8) but it suffers in recall and f1-score. Our proposed *DeepCausalEvent* model achieves the best recall and f1-scores (results are shown in bold in Table 8) among the deep learning approaches as we apply sequence-aware event representation technique, an event context word extension technique with a deep neural networks model. We also find that in the *DeepCausalEvent* model, Word2vec word embedding achieves better performance compared to the BERT embedding, which shows that Word2vec embedding encodes more contextual features than the BERT word embeddings.

5.4 | Discussion

Table 9 compares the prediction of a few examples of candidate causal pairs. The table displays a set of cause-effect event pairs with their annotated labels. It also displays the predicted labels of the pairs by the benchmark approaches and our proposed *DeepCausalEvent* model. From the table, we can see that our previous work *FFNN+2-word Ext*. performs batter then the other benchmark approaches. For example, *samoa's don opeloge lifts 191kg* → *he wins* and *he does not know beyond cricket* → *homework neded* is a causal event pair but only detected by our previous work *FFNN+2-word Ext*. and *DeepCausalEvent*. Comparing our proposed model with the previous model *FFNN+2-word Ext*., we find that the similar results for every examples except two cases. For example, *imoral atack on syrian childrens* → *cni trump should be impeached and hanged til death* is not a causal event but *FFNN+2-word Ext*., mistakenly identifies it as causal whereas our proposed model *DeepCausalEvent* identifies it as noncausal. On the other hand, *the task of carying your country's flag embodies the values and ideals it represents* → *we have found a perfect role model for its cause* is a causal event pair that *FFNN + 2-word Ext*. cannot detect it but *DeepCausalEvent* can identifies it as causal. There is an example: *#cameronvanderburgh* → *big upset at #comonwealthgames* for which all the models in the experiment predicted the wrong label. This particular case represents the challenge of detecting causality in tweets when words are written as hashtags.

6 | CONCLUSION

In this paper, we propose an event causality detection model for tweets that mines and detects causally related events by applying a causal background knowledge network and a deep neural network model. We find that the proposed event context word extension technique contributes to enhance the feature sets for the neural network models. We also find that our sequence aware event representation and deep neural network-based model improves the performance of event causality detection in tweets. In our experiments, we notice the improved performance of deep neural network-based models when the model is trained on enhanced feature set. Our proposed model can be used by event management authorities in a smart city to timely detect causally related events and take appropriate actions when necessary.

ORCID

Humayun Kayesh https://orcid.org/0000-0002-9975-5862
Md. Saiful Islam https://orcid.org/0000-0001-7181-5328
Junhu Wang https://orcid.org/0000-0003-2962-1604
A.S.M. Kayes https://orcid.org/0000-0002-2421-2214
Paul A. Watters https://orcid.org/0000-0002-1399-7175

REFERENCES

1. Preoţiuc-Pietro D, Liu Y, Hopkins D, Ungar L. Beyond binary labels: political ideology prediction of twitter users. Paper presented at: Proceedings of the Annual Meeting of the Association for Computational Linguistics; 2017:729-740.
2. Hasanuzzaman M, Kamila S, Kaur M, Saha S, Ekbal A. Temporal orientation of tweets for predicting income of users. Paper presented at: Proceedings of the Annual Meeting of the Association for Computational Linguistics; 2017:659-665.
3. Liu Y, Zhang L, Nie L, Yan Y, Rosenblum DS. Fortune teller: predicting your career path. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Association for the Advancement of Artificial Intelligence; 2016;201-207.
4. Kayesh H, Islam MS, Wang J. A causality driven approach to adverse drug reactions detection in tweets. Paper presented at: Proceedings of the International Conference on Advanced Data Mining and Applications (ADMA); 2019:316-330.
5. Kayesh H, Islam MS, Wang J, Anirban S, Kayes A, Watters P. Answering binary causal questions: a transfer learning based approach. Paper presented at: Proceedings International Joint Conference on Neural Networks (IJCNN); 2020.
6. Cocos A, Fiks AG, Masino AJ. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts. *JAMIA*. 2017;24(4):813-821.
7. Kayesh H, Islam MS, Wang J. Event causality detection in tweets by context word extension and neural networks. Paper presented at: Proceedings of the International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT); 2019:352-357.
8. Blanco E, Castell N, Moldovan D. Causal relation extraction. Paper presented at: Proceedings of the International Conference on Language Resources and Evaluation (LREC); 2008:310-313.
9. Do QX, Chan YS, Roth D. Minimally supervised event causality identification. Paper presented at: Proceedings of the EMNLP; 2011:294-303.
10. Ciresan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. Paper presented at: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR); 2012:3642-3649.
11. Riaz M, Girju R. Toward a better understanding of causality between verbal events: extraction and analysis of the causal power of verb-verb associations. Paper presented at: Proceedings of the SIGDIAL; 2013:21-30.
12. Riaz M, Girju R. Recognizing causality in verb-noun pairs via noun and verb semantics. *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (EACL-CAtoCL)*. Gothenburg, Sweden: Association for Computational Linguistics; 2014;48-57.
13. Mirza P. Extracting temporal and causal relations between events. Paper presented at: Proceedings of the ACL Student Research Workshop; 2014:10-17.
14. Luo Z, Sha Y, Zhu KQ, Hwang SW, Wang Z. Commonsense causal reasoning between short texts. Paper presented at: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR); 2016:421-431.

15. Kruengkrai C, Torisawa K, Hashimoto C, Kloetzer J, Oh JH, Tanaka M. Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Association for the Advancement of Artificial Intelligence; 2017;3466-3473.
16. Ritter A, Clark S, Etzioni O. Named entity recognition in tweets: an experimental study. Paper presented at: Proceedings of the EMNLP; 2011:1524-1534.
17. Sasaki S, Takase S, Inoue N, Okazaki N, Inui K. Handling multiword expressions in causality estimation. Paper presented at: Proceedings of the IWCS; 2017.
18. Doan S, Yang EW, Tilak SS, Li PW, Zisook DS, Torii M. Extracting health-related causality from twitter messages using natural language processing. *BMC Med Inf Decis Making*. 2019;19-S(3):71-77.
19. Ponti EM, Korhonen A. Event-related features in feed forward neural networks contribute to identifying causal relations in discourse. *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*. Valencia, Spain: Association for Computational Linguistics; 2017;25-30.
20. Kayesh H, Islam MS, Wang J. On event causality detection in tweets; 2019. arXiv preprint arXiv:1901.03526.
21. Yu B, Li Y, Wang J. Detecting causal language use in science findings. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics; 2019;4663-4673.
22. Prasad R, Miltsakaki E, Dinesh N, et al. *The Penn Discourse Treebank 2.0 Annotation Manual*. IRCS Technical Reports Series; 2007:203.
23. Chambers N, Jurafsky D. Unsupervised learning of narrative event chains. *Proceedings of ACL-08: HLT (ACL: HLT)*. Columbus, OH: Association for Computational Linguistics; 2008;789-797.
24. Turney PD, Pantel P. From frequency to meaning: vector space models of semantics. *JAIR*. 2010;37:141-188.
25. Rahimtoroghi E, Hernandez E, Walker MA. Learning fine-grained knowledge about contingent relations between everyday events. Paper presented at: Proceedings of the SIGDIAL 2016:350-359.
26. Khan S, Parkinson S. Causal connections mining within security event logs. *Proceedings of the International Conference on Knowledge Capture (K-CAP)*. Austin, TX: Association for Computing Machinery; 2017;38.38:1–38:4.
27. Roberts K, Harabagiu SM. Detecting new and emerging events in streaming news documents. *Int J Semant Comput*. 2011;5(4):407-431.
28. Mirza P. Extracting temporal and causal relations between events. *Proceedings of the ACL 2014 Student Research Workshop*. Baltimore, MD: Association for Computational Linguistics; 2014; abs/1604.08120.10–17.
29. Balashankar A, Chakraborty S, Fraiberger S, Subramanian L. Identifying predictive causal factors from news streams. In: Inui K, Jiang J, Ng V, Wan X., eds. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics; ; 2019: 2338–2348.
30. Chen D, Manning C. A fast and accurate dependency parser using neural networks. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics; 2014;740-750.
31. Radinsky K, Davidovich S, Markovitch S. Learning causality for news events prediction. Paper presented at: Proceedings of the WWW Conference; 2012: 909–918.
32. Jang B, Yoon J. Characteristics analysis of data from news and social network services. *IEEE Access*. 2018;6:18061-18073.
33. Hassanzadeh O, Bhattacharjya D, Feblowitz M, et al. Answering binary causal questions through large-scale text mining: an evaluation using cause effect pairs from human experts. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*. Macao, China: ijcai.org; 2019;5003-5009.
34. Corney D, Albakour D, Martinez M, Moussa S. What do a million news articles look like? *Proceedings of the NewsIR'16 Workshop (NewsIR)*. Padua, Italy: CEUR-WS.org; 2016;42-47.
35. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*. Lake Tahoe, NV: Neural Information Processing Systems Foundation; 2013;2 3111-3119.

How to cite this article: . A deep learning model for mining and detecting causally related events in tweets. *Concurrency Computat Pract Exper*. 2020;e5938. https://doi.org/10.1002/cpe.5938