

Event Causality Detection in Tweets by Context Word Extension and Neural Networks

Humayun Kayesh, Md. Saiful Islam, and Junhu Wang
School of ICT, Griffith University, Australia
 humayun.kayesh@griffithuni.edu.au, {saiful.islam, j.wang}@griffith.edu.au

Abstract—Twitter has become a great source of user-generated information about events. Very often people report causal relationships between events in their tweets. Automatic detection of causality information in these events might play an important role in prescriptive event analytics. Existing approaches include both rule-based and data-driven supervised methods. However, it is challenging to identify event causality accurately using linguistic rules due to the unstructured nature and grammatical incorrectness of social media short text such as tweets. Also, it is difficult to develop a data-driven supervised method for event causality detection in tweets due to insufficient contextual information. This paper proposes a novel event context word extension technique based on background knowledge. To demonstrate the effectiveness of our event context word extension technique, we develop a feed-forward neural network based approach to detect event causality from tweets. Extensive experiments demonstrate the superiority of our approach.

Index Terms—Event Causality, Feed-Forward Neural Network, Feature Enhancement

I. INTRODUCTION

Microblogging sites such as Twitter have become a popular medium for users to express their opinion and respond to different situations. Therefore, tweets can be an important source of causality information between events and this information might play an important role in prescriptive event analytics. For example, the following tweet “*A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers*” contains two causally related events. From this tweet it can be said that the “lack of communication” was a cause of transport service disruption in Gold Coast. This information could be applied in prescriptive analytics by the decision makers to reduce the chance of a future transport disruption during public gatherings. Causality information can also be applied to improve automated *why* question answering. For example, we can answer the question “Why is Sally Pearson not participating today?” from the event causality information extracted from the following tweet, “*A knee injury caused Sally Pearson to quit the competition*”. The uses of the above event causalities in the prescriptive event analytics based applications are visualized in Fig. 1.

The extraction of causal relationships is an evolving area of research [1]–[6]. Existing approaches often apply linguistic rules or commonsense knowledge to identify causal relationship from short text. However, processing tweets is more challenging than normal short text. One of the most obvious challenges is that tweets are unstructured and highly informal in nature. Hence, the linguistic rule-based approaches [5], [7],

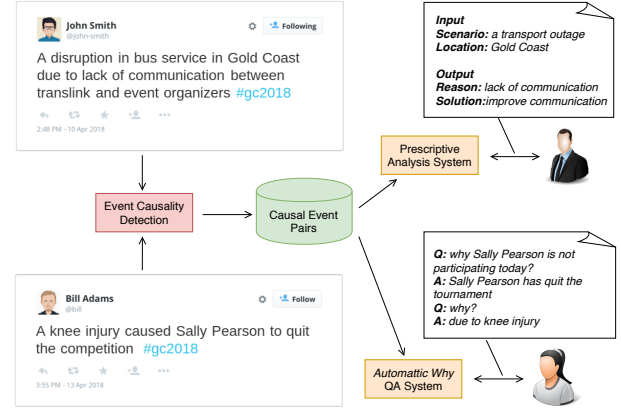


Fig. 1: Application of automatic event causality detection in *why* question answering and prescriptive event analysis

[8], which depend on grammatical correctness of text, perform poorly on tweets (see section V). Additionally, the causally related event pairs appear infrequently in tweets and the causal relationship between events lacks context information. Due to this lack of adequate context information, supervised learning based approaches such as [9] are not much effective on tweets. In this paper, we propose an automated approach to detect causally related event pairs from social media short text, e.g. tweets. In our proposal, we use a feed-forward neural network to detect causal relationship between events to deal with the unstructured nature of the data. To accurately train the model, we propose a context word extension method to enhance the feature set. We extend the event keywords of both candidate causal event and candidate effect event using background knowledge before applying the model. The background knowledge is captured by creating a causal network from news article text using a set of causal cue words. To be specific, the main contributions of this paper are as follows:

- 1) we propose a novel event context word extension technique that uses commonsense background knowledge extracted from news articles to enhance the feature set;
- 2) we develop a neural network based event causality detection method to detect causal relationship between events;
- 3) we perform an extensive experimental evaluation to demonstrate that our proposed event context word extension technique outperforms the method of using only the event keyword and other attribute words; and
- 4) we also compare the performance of our method with existing methods of causality detection from short text.

II. RELATED WORK

Phrase-based Causality Detection. One of the early approaches that identifies explicit, implicit and non-causal relationships between verb-verb pairs is proposed in [3]. Later, in [4] the authors extend the previous approach and extract causal relationship between noun-verb pairs. The authors at first identify all the nouns and verbs from a sentence and then apply a supervised classifier to identify causality between grammatically connected noun-verb pairs. A more recent approach [5] builds a causality network of terms from a collection of web text. The authors apply linguistic rules, e.g., ‘A causes B’ to extract the terms with causal relations. This approach is extended by Sasaki et al. [8] for multi-word terms where authors calculated causal strength not only for the pairs of single words but also for the multi-word pairs. Oh et al. [10] proposed a CNN based approach for automatic *why* question answering using causality. Although the above approaches calculate causal strengths between phrases, they do not take events into account.

Event-based Causality Detection. Some approaches extract causal relationships between event pairs. Do et al. [1] propose an event causality detection method that uses distributional probability and discourse connectives. The authors calculate *pointwise mutual information* (PMI) [11] scores between different elements of event pairs. However, the PMI based approaches are sensitive to co-occurrence frequency and do not perform well for infrequent events [12]. Radinsky et al. [7] propose an automatic rule generation technique to detect causality between news events. However, the rule-based approaches are more appropriate for the formal text such as new headlines rather than tweets.

Recently, Kruengkrai et al. [6] propose a method that utilizes background knowledge to determine the causal relationship between two candidate events. The authors apply a *multi-column neural network* [2] to extract causal relationships between candidate phrases using archived web text. However, this approach does not consider the spatial and temporal information of events. An event causality detection approach that is closely related to our approach is proposed in [9] which uses a feed-forward neural network for detecting causality relation between events. They enhance the feature set by calculating distances between event trigger word and other words in the sentence. However, this positional information might not represent the causal strength properly as tweets often contain noisy characters and words e.g., emojis, hashtags and mentions and therefore, may not be applicable for event causality detection in tweets.

III. PROBLEM FORMULATION

An event is modeled as a set of words that represents the occurrence of a specific incident. The event keyword is the word that triggers the event and the event attributes are the words that are syntactically related to the event keyword. Table I displays a few example events¹. An event is considered

TABLE I: Representation of Events

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

as *causal* if the event causes another event to occur. The other event is considered as the *effect* of the causal event. For Example, “**A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers**” contains a causal event and an effect event. In the above example the causal event is ‘lack of communication’ and the effect event is ‘a disruption in bus service’, where *disruption* is the event keyword of the causal event and *lack* is the event keyword of the effect event. This paper focuses on extracting such causal and effect event pairs from tweets and aims to address the following research question (RQ).

RQ: *How to automatically detect causally related event pairs from social media short texts such as tweets?*

We assume that causal and effect events occur in the same sentence in a tweet. We aim to detect such explicitly mentioned causal and effect event pairs. We define a candidate causal event as e_1 and effect event as e_2 . Each event has a structure of $k(a_1, a_2, a_3, \dots, a_n)$, where k is the event keyword or trigger word and $\{a_i\}$ are the event attributes. For a candidate causal event pair (e_1, e_2) , our goal is to classify if the event pair has a causal relationship between them, i.e., e_1 causes e_2 ². Formally, we define the problem studied in this paper as follows:

$$f(e_1, e_2) = \begin{cases} \text{Causal} & \text{if } e_1 \text{ causes } e_2, \\ \text{Not Causal} & \text{otherwise} \end{cases} \quad (1)$$

where f is a function that takes two events e_1 and e_2 as input and outputs either ‘Causal’ or ‘Not Causal’. The function outputs ‘Causal’ if e_1 causes e_2 in the input event pair and it outputs ‘Not Causal’ otherwise.

IV. OUR APPROACH

Our proposed method utilizes background knowledge to detect event causality. The background knowledge is extracted from news articles in the form of a causal network. To apply background knowledge, we extend event context words using the causal network. The events are then converted into word vectors to train a feed-forward neural network. The trained model is then used to detect causal relationship between a new pair of candidate causal events.

A. Tweet Preprocessing

As a first step of preprocessing, tweets are split into sentences. Sentences in tweets often contain characters that are considered as noises such as emojis, repeated characters and symbols. We perform a series of preprocessing steps to remove noisy characters from sentences. These steps include removal of non-alphabetic characters such as emojis, symbols, hashtags (#) and mention (@) characters, and URLs. We

¹It should be noted that a sentence may have zero or more events.

²Although e_1 causes e_2 , it does not mean that e_1 is the only cause of e_2 . There could be other causes of e_2 , which is beyond the scope of this paper.

TABLE II: Causal Cue Words

affect	because	causes	due to	if	induce	owing to	results from
affected by	because of	causing	effect of	if..., then	induced	reason for	so that
affects	bring on	consequently	for this reason alone	in consequence of	inducing	reason of	that's why
and consequently	brings on	coz	gave rise to	in response to	lead to	reasons for	the result is
and hence	brought on	coz of	give rise to	inasmuch as	leading to	reasons of	thereby
as a consequence	cause	decrease	given rise to	increase	leads to	result from	therefor
as a consequence of	caused	decreased by	giving rise to	increased by	led to	resulted from	thus
as a result of	caused by	decreases	hence	increases	on account of	resulting from	

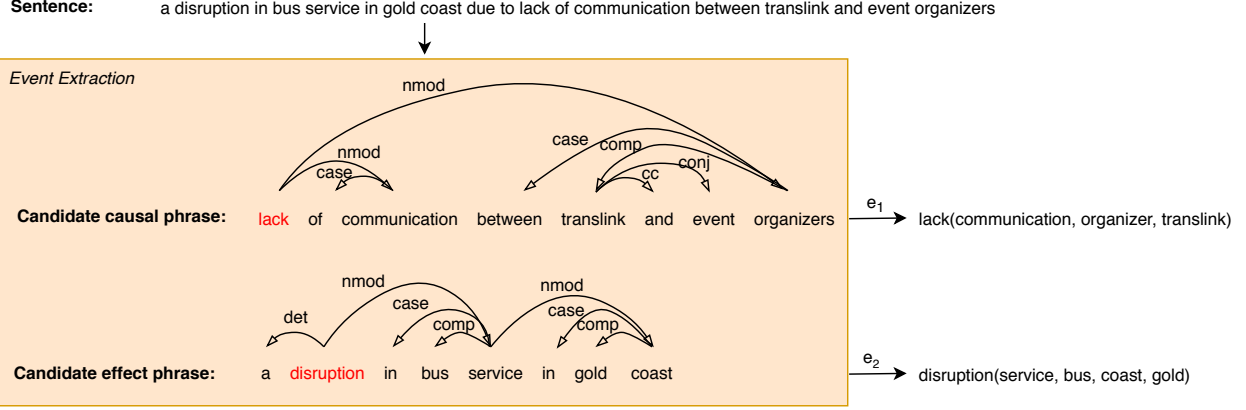


Fig. 2: An example of event pair extraction from a sentence

also discard sentences ending with question mark (“?”) and normalize repeated characters in a word, e.g., ‘yesss’ to ‘yes’.

B. Event Pair Extraction

In this step, a pair of candidate events is extracted from a sentence. At first, a sentence is split into candidate causal and effect phrases using a set of causal cue words [5], [13] (please see Table II). For example, “a disruption in bus service in gold coast due to lack of communication between translink and event organizers” is split into “a disruption in bus service in gold coast” as the candidate effect phrase and “lack of communication between translink and event organizers” as the candidate causal phrase using the cue word *due to*. Candidate cause and effect phrases are then passed to the Stanford dependency parser [14] to detect the root word for each phrase that are considered to be the event keyword k of the corresponding events e_1 (or e_2). The other words that are related to the root word via ‘nsubj’, ‘nsubjpass’, ‘amod’, ‘dobj’, ‘advmod’, ‘nmod’, ‘xcomp’, ‘compound:prt’, ‘compound’ and ‘neg’ relationships are extracted as the event attributes $\{a_i\}$. We also extract words that are related to the extracted event attributes via the above relationships as the event attributes. An example of event keyword and attributes extraction is illustrated in Fig. 2.

C. Causal Network

Background knowledge plays an important role in event causality detection. We use 1 million news articles³ from the work of [15] as a source of background knowledge and store the captured knowledge as causal relationships in a network called *causal network*. To construct the network, first we extract the causal and effect phrases from article sentences

using the causal cue words given in Table II. Then, the phrases are converted to lower cases after removing the stop words. The phrases are then tokenized and lemmatized. Each token in the either phrase represents a node in the network. A directed link from token A to token B contains frequency such that token A appeared in a causal phrase and token B appeared in the corresponding effect phrase, which is illustrated in Fig. 3.

D. Context Word Extension

We use the background knowledge captured in the causal network to extend candidate event context words. For example, if there is a causal relationship between ‘Rain’ and ‘Flood’, it can be said that in many previous occasions the word *Rain* was part of causal phrases where *Flood* was a part of effect phrases. This knowledge can be applied to the causality detection method for extending context words. In our approach, we look for the corresponding effect event keyword in the causal network to extend the context word k of a candidate causal event. First, we identify a list of words with inward links to effect keyword in causal network. The list is then sorted in descending order of their frequencies. From that sorted list, we take the top n words to extend the context of candidate causal event e_1 , where n is the number of words we want to extend. Similarly, to extend the context of candidate effect event e_2 , we identify the top n effect words from causal network. A running example is given in Fig. 4 to illustrate our context word extension technique for *lack* and *disruption* event keywords using the causal network.

E. Feature Extraction

In the feature extraction stage, the candidate causal event e_1 and the candidate effect event e_2 are converted into a numerical vector v . However, before the conversion, the context words of e_1 and e_2 are extended following the steps described in Section

³<https://research.signalmedia.co/newsir16/signal-dataset.html>

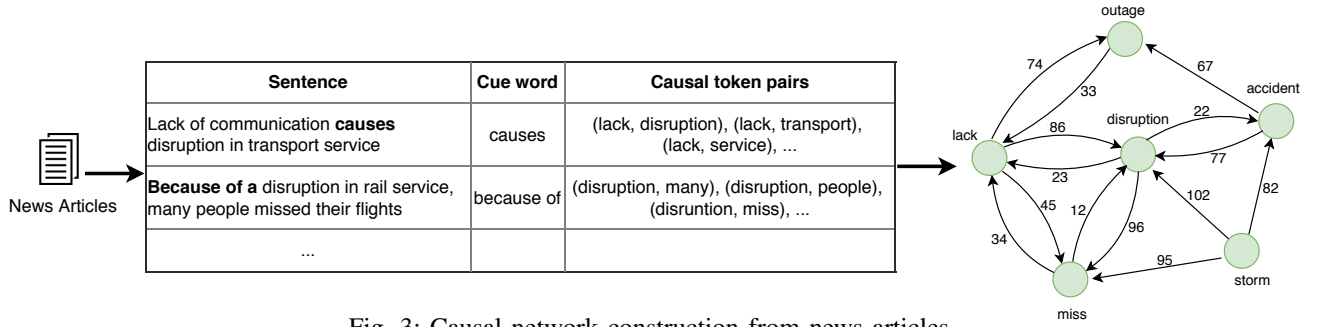


Fig. 3: Causal network construction from news articles

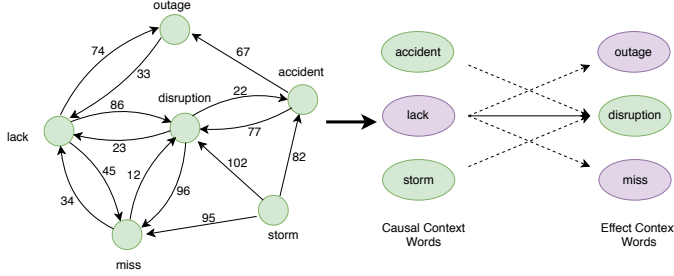


Fig. 4: An example of n -word context word extension, where $n = 2$ and the original candidate cause and effect keywords are *lack* and *disruption*, respectively

IV-D, which generates e'_1 and e'_2 respectively. To convert e'_1 and e'_2 into v , we train a Word2vec model [16] from 1 million news articles (the same dataset that is used to build the *causal network*). Then we extract the dictionary of words \mathcal{D} from the trained Word2vec model. Using this dictionary we replace every word in e'_1 and e'_2 by its corresponding index in \mathcal{D} . The word indexes of e'_1 and e'_2 are then concatenated together to construct a single index vector i_v . In the next step, each index i_v is replaced by its corresponding word embedding which produces a matrix of word embeddings M . The number of columns in M is 300 and the number of rows is the same as the total words e'_1 and e'_2 . Finally, the matrix M is flattened by taking mean and converted to a vector v of size 300. This vector v is passed to the input layer of the feed-forward neural network for training and detection, which is discussed below.

1) *Learning the Detection Model*: In this step, we train a feed-forward neural network model. First, we prepare a gold standard dataset that contains event pairs where each pair is labeled as either ‘causal’ or ‘not causal’. Then we extract feature for the candidate event pair following the steps described in IV-E, which include context word extension and vectorization. In the context word extension step, we extend the event context word k for both causal event e_1 and effect event e_2 using a pre-constructed causal network (see Sec. IV-D). The context word extension step generates e'_1 and e'_2 where e'_1 is the extended version of e_1 and e'_2 is the extended version of e_2 . After performing the context word extension, every event pair is converted into a 300 dimensional feature vector following the steps described in Sec. IV-E. Such feature vectors of all candidate event pairs and their corresponding labels (‘causal’ or ‘not causal’) are passed to a feed-forward neural network for training. The trained model is then used to

detect the causal relationship between candidate event pairs in unknown tweets.

2) *Causal Relationship Detection*: The causal relationship detection between event pairs in an unknown tweet starts with a series of preprocessing steps as described in Section IV-A. After preprocessing, the pairs of candidate causal events are extracted (please see Section IV-B). The next step is to extract features, where context word extension technique is applied to both candidate causal event and effect event (see Section IV-D). The event words are then converted into the feature vector (see Section IV-E) which is passed to the trained feed-forward neural network model for event causality detection.

The schematic diagram of causal relationship detection in candidate event pairs, which includes learning the neural network model as well as the detection of causal relationship in unknown event pairs, is illustrated in Fig. 5.

V. EXPERIMENTS

This section presents our experiments and demonstrates the effectiveness of our method on event causality detection.

Dataset. We collect 207,705 tweets that are related to the Commonwealth Games 2018 in Gold Coast, Australia and posted during the period from 2017-10-05 to 2018-05-07 using twitter API⁴. The following hashtags are used as keywords to collect the tweets: ‘#CommonwealthGames’, ‘#CommonwealthGames2018’, ‘#GC2018’, and ‘#ShareTheDream’. After performing the preprocessing steps mentioned in Section IV-A, we identify 913 candidate cause and effect event pairs based on the approach described in Section IV-B and annotate them manually as either ‘Causal’ or ‘Not Causal’. We split our annotated dataset to separate 60% data for training and 40% for testing. We ensure that the ratio of ‘Causal’ and ‘Not Causal’ data remains same in both training and test data. The statistics of the tested dataset is presented in Table III. Among the training data (60% of the original dataset), we use 50% data for learning the model and the rest 50% data for validation and parameter optimization.

Setup. We implement the proposed method in Python 3.6 and use *keras* python package to implement the feed-forward neural network based causality detection method. The neural network has an

TABLE III: Dataset Statistics

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

⁴<https://developer.twitter.com/en/docs/tweets/search/overview>

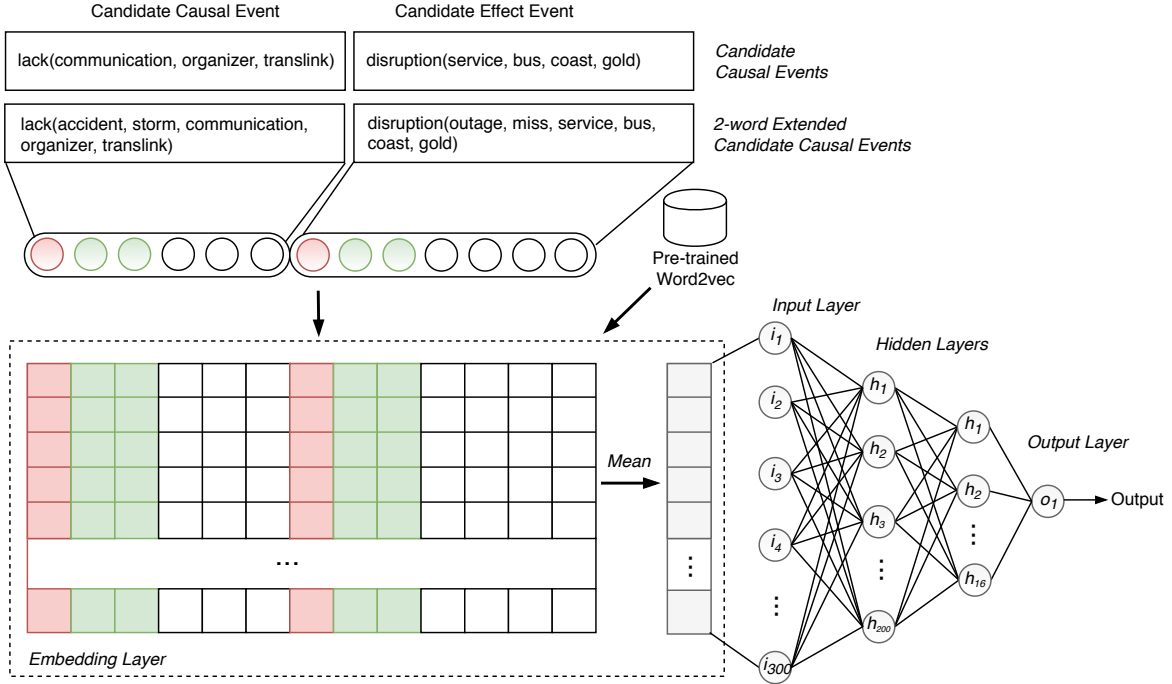


Fig. 5: Feature extraction from an event pair and different layers of the feed-forward neural network

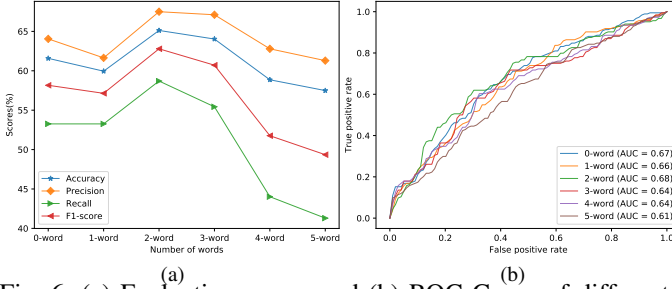


Fig. 6: (a) Evaluation scores and (b) ROC Curve of different settings in the proposed context word extension technique

input layer, two hidden layers and an output layer. The input layer contains 300 nodes and we use ‘ReLU’ as the activation function, which accepts 300 dimensional event vectors as the input. Next to this layer we have two fully connected hidden layers that consists of 200 nodes and 16 nodes, respectively (this topology has been optimized empirically). Each node in both of the hidden layers uses ‘ReLU’ as the activation function. The output layer is another fully connected layer that has only one node with an activation function. We use ADADELTA as the cost function with batch size 40 and evaluation metric: “Accuracy”, to optimize the parameters.

Parameter Optimization. We optimize the learning rate of our feed-forward neural network model while keeping other parameters fixed. Initially, we set our learning rate to 0.01 and raise the learning rate gradually. We find the best learning performance for learning rate 0.1. We perform this experiment for 0 to 5 event context word extensions and run for 250 iterations. We observe that the validation accuracy stops growing or starts to decline after 150 iterations.

Performance Evaluation. To evaluate the generalizability of our proposed method of event causality detection, we

TABLE IV: Comparison of the proposed method with existing approaches

Methods	Accuracy	Precision	Recall	F1-score
Commonsense [5]	50.95	56.67	9.24	15.89
Commonsense + Multi-word [8]	50.14	54.55	3.26	6.15
FFNN + Position [9]	59.40	60.12	56.52	58.26
FFNN + 2-word Extension (ours)	65.94	67.46	61.96	64.59

perform experiment on different settings of event context word extension. Fig. 6 (a) illustrates the standard evaluation scores: accuracy, precision, recall and F1-score of different experiment settings. The results suggest that we gain performance improvement across the evaluation scores for both 2 and 3-word extension compared to the model that uses 0-word extension. The model with 2-word extension achieves the best evaluation scores. We also observe that increasing the number of word extension after 3 does not perform well and the performance drops sharply. This is because the words extracted from the knowledge base become more prominent than the original event words. From Fig. 6 (b), we see that the Area Under Curve (AUC) value for 2-word extension based method is higher than the 0-word extension based method. In conclusion, the 2-word extension based method is the best performer.

To compare the performance of the proposed method with other existing methods, we implement three existing causality detection approaches. We implement the commonsense causality detection method (Commonsense) proposed by Luo et al. [5]. Additionally, we implement another approach proposed by Sasaki et al. [8], which is denoted as ‘Commonsense + Multi-word’. The third existing method is an event causality detection approach [9] (FFNN + Position). We set iterations to 150, learning rate to 0.1 and batch size to 1 to train the neural network for this method. From Table IV, it is evident that the

TABLE V: Some examples of candidate causal pairs (causal \rightarrow effect) and their predicted labels by different methods. The label ‘1’ and ‘0’ represent ‘Causal’ and ‘Not Causal’ relations respectively. The column ‘Gold Data’ shows the ground truth.

Candidate Causal Pairs	Gold Data	Commonsense [5]	Commonsense + Multi-word [8]	FFNN + Position [9]	FFNN + 2-word Extension (ours)
persistent achilles injury \rightarrow disapointed @salypearson won't be runing at #gc2018 #comonwealthgames	1	0	0	1	1
samoa's don opeloge lifts 191kg \rightarrow he wins	1	0	0	0	1
no tickets \rightarrow babita's father mised her #comonwealthgames2018-match	1	0	0	1	1
imoral atack on syrian childrens \rightarrow @cni trump should be impeached and hanged til death	0	0	0	1	1
her father's name is not cleared as "an oficial" \rightarrow she wil not take part in #gc2018	1	0	0	1	1
presure from the defence \rightarrow a lose pas from malawi	1	0	0	1	1
i want to watch it al live \rightarrow is there any legislation i can use to work from home until the #comonwealthgames2018 finishes	0	1	0	1	0
you're in the area \rightarrow please be aware there wil also be road closures and parking restrictions on competition days on 8	0	1	0	0	1
a technical issue \rightarrow 34am central to varsity lakes train is delayed 30 minutes	1	1	1	1	1
#cameronvanderburgh \rightarrow big upset at #comonwealthgames	1	0	0	0	0

performance gain achieved by our method over the competitor methods is at least 9.6% and 10.87% in terms of Recall and F1-score, respectively. These outcomes demonstrate that our event context word extension technique is capable of overcoming the issue of insufficient context information in candidate causal event pairs in tweets.

Discussion. The key findings of this research is that the overall performance of event causality detection in tweets can be improved significantly by combining commonsense knowledge with neural network. The recall values of *Commonsense* [5] and *Commonsense + Multi-word* [8] in Table IV are low because they solely depend on the causal co-occurrence scores of words in the causal network and ignore the semantic relationship between words. In contrast, the neural network based approaches outperform commonsense-based methods significantly. This is because tweets often contain real life causal events e.g., *her father's name is not cleared as "an oficial" \rightarrow she wil not take part in #gc2018*, which is difficult to explain using commonsense. In such cases the neural network-based methods are more effective than the commonsense-based methods due to their capability to learn from context rather then applying limited commonsense information. Furthermore, the proposed context word extension based feature enhancement technique improves the event causality detection capability of neural network better compared to the positional information of words in a tweet. Table V demonstrates the effectiveness of our proposed context word extension-based method in situations where context information is limited. For example, every method, except the proposed method, used in the experiment fails to detect the causal relationship for the causal pair *samoa's don opeloge lifts 191kg \rightarrow he wins*. In this case, our approach is able to take advantage of the context word extension technique to enhance the detection capability of the neural network. However, the proposed method does have some limitations. Sometimes, it incorrectly detects causal events pairs opinions, suggestions or sarcastic statements. For example, *#cameronvanderburgh \rightarrow big upset at #comonwealthgames* as shown in Table V that is undetected by every method. This particular example showcases another challenge of event causality detection in tweets, where the causal event is written as a hashtag and without any space. Extracting causal features from such tweets

can be an important research direction.

VI. CONCLUSION

This paper proposes a feature enhancement technique for supervised learning based event causality detection approach. We demonstrate that commonsense background knowledge can be used to extend event context information, which helps to enhance feature set of a supervised learning based method. We develop a neural network based event causality detection method that uses event context word extension technique to detect causal relationship between pair of events. We find that the neural network based model performs better when trained on the enhanced feature set for the causal relationship detection in tweets.

REFERENCES

- [1] Q. X. Do, Y. S. Chan, and D. Roth, "Minimally Supervised Event Causality Identification," *EMNLP*, pp. 294–303, 2011.
- [2] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *CVPR*, pp. 3642–3649, 2012.
- [3] M. Riaz and R. Girju, "Toward a Better Understanding of Causality between Verbal Events: Extraction and Analysis of the Causal Power of Verb-Verb Associations," *SIGDIAL*, pp. 21–30, 2013.
- [4] M. Riaz and R. Girju, "Recognizing Causality in Verb-Noun Pairs via Noun and Verb Semantics," *EACL-CAtoCL*, pp. 48–57, 2014.
- [5] Z. Luo, Y. Sha, K. Q. Zhu, S.-w. Hwang, and Z. Wang, "Commonsense Causal Reasoning between Short Texts," *KR*, pp. 421–431, 2016.
- [6] C. Kruegkrai, K. Torisawa, C. Hashimoto, J. Kloetzer, J.-H. Oh, and M. Tanaka, "Improving Event Causality Recognition with Multiple Background Knowledge Sources Using Multi-Column Convolutional Neural Networks," *AAAI*, 2017.
- [7] K. Radinsky, S. Davidovich, and S. Markovitch, "Learning causality for news events prediction," in *WWW*, pp. 909–918, ACM, 2012.
- [8] S. Sasaki, S. Takase, N. Inoue, N. Okazaki, and K. Inui, "Handling Multiword Expressions in Causality Estimation," *IWCS*, 2017.
- [9] E. M. Ponti and A. Korhonen, "Event-related features in feed forward neural networks contribute to identifying causal relations in discourse," *LSDSem*, pp. 25–30, 2017.
- [10] J.-H. Oh, K. Torisawa, C. Kruegkrai, R. Iida, and J. Kloetzer, "Multi-column convolutional neural networks with causality-attention for why-question answering," in *WSDM*, pp. 415–424, ACM, 2017.
- [11] N. Chambers and D. Jurafsky, "Unsupervised Learning of Narrative Event Chains," *ACL: HLT*, pp. 789–797, 2008.
- [12] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *JAIR*, vol. 37, pp. 141–188, 2010.
- [13] P. Mirza, "Extracting Temporal and Causal Relations between Events," *ACL Student Research Workshop*, pp. 10–17, 2014.
- [14] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," *EMNLP*, pp. 740–750, 2014.
- [15] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?," in *NewsIR*, pp. 42–47, 2016.
- [16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, pp. 3111–3119, 2013.