Deep Learning for Causal Discovery in Texts

Deep Learning for Causal Discovery in Texts

by

Humayun Kayesh

B.Sc., M.Sc.

School of Information and Communication Technology

Griffith University, Gold Coast, Australia

Submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

March 2022

ABSTRACT

Causality detection in text data is a challenging natural language processing task. This is a trivial task for human beings as they acquire vast background knowledge throughout their lifetime. For example, a human knows from their experience that *heavy rain* may cause *flood* or *plane accidents* may cause *death*. However, it is challenging to automatically detect such causal relationships in texts due to the availability of limited contextual information and the unstructured nature of texts. The task is even more challenging for social media short texts such as Tweets as often they are informal, short, and grammatically incorrect.

Generating hand-crafted linguistic rules is an option but is not always effective to detect causal relationships in text because they are rigid and require grammatically correct sentences. Also, the rules are often domain-specific and not always portable to another domain. Therefore, supervised learning techniques are more appropriate in the above scenario. Traditional machine learning-based model also suffers from the high dimensional features of texts. This is why deep learning-based approaches are becoming increasingly popular for natural language processing tasks such as causality detection. However, deep learning models often require large datasets with high-quality features to perform well. Extracting deeply-learnable causal features and applying them to a carefully designed deep learning model is important. Also, preparing a large human-labeled training dataset is expensive and time-consuming. Even if a large training dataset is available, it is computationally expensive to train a deep learning model due to the complex structure of neural networks. We focus on addressing the following challenges: (i) extracting high-quality causal features, (ii) designing an effective deep learning model to learn from the causal features, and (iii) reducing the dependency on large training datasets.

Our main goals in this thesis are as follows: (i) we aim to study the different aspects of causality and causal discovery in text in depth. (ii) We aim to develop strategies to model causality in text, (iii) and finally, we aim to develop frameworks to design effective and efficient deep neural network structures to discover causality in texts.

STATEMENT OF ORIGINALITY

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Humayun Kayesh

June 10, 2022

ACKNOWLEDGMENTS

I would like to express my gratitude to almighty Allah for keeping me safe, sound, and healthy throughout the time of my PhD. I am also grateful to him for giving me the strength and courage to overcome all the struggles and challenges on the way to completing my PhD.

I would like to thank my principal supervisor A/Prof. Junhu Wang for his continuous help and support since the day I submitted my PhD application to Griffith. This was such a great experience to work under his supervision. Despite being busy with courses and research work, he met me regularly to discuss my progress and gave me directions. The regular meetings with him kept my research on track. His thoughtful feedback improved the quality of my work.

I cannot just thank enough my co-principal supervisor, Dr. Saiful Islam, for his constant support and guidance. He is one of the best teachers, supervisors, and mentors I have ever worked with. He was by my side during all the ups and downs of my PhD life. He helped me to cope with the difficult times when papers got rejected or experiments were not working as expected. His rich knowledge, expertise, and endless energy always encouraged me to push harder and bring out the best out of myself. I am so grateful that he believed in me and kept confidence in me till the end of my PhD.

I would like to pass my heartfelt thanks to my wife Mushsarat Jahan for taking care of me so that I can focus on my study. She has been very supportive throughout my PhD and her presence instilled positive energy into my life every day. I am also blessed to have my daughter Parisa Kayesh with me. She was born during my PhD, and since then she has been my biggest source of comfort and happiness. Playing with her refreshed my mind and recharged my energy when I was stressed out from deadlines and workloads.

I am grateful to my parents without whom I wouldn't be able to accomplish anything in my life. My father was one of the biggest well-wishers of me but unfortunately, he died from Covid-19 during my PhD. I am lucky to have such a mother who prays for me every day from Bangladesh. I respectfully acknowledge my parents' sacrifice to educate me in

my childhood despite their financial hardship. I am also grateful to my parents-in-law for their moral support and encouragement. I remember the support and encouragement from my elder brothers, younger sister, brothers-in-law, and all other family members.

I am thankful to Griffith University for granting me the scholarship and giving me the opportunity to complete my PhD. At the school of Information and Communication Technology, I had everything I needed to conduct my research successfully. I remember Prof. Goran Nenadic who was my MSc thesis supervisor at the University of Manchester. He was a great supervisor and I learned a lot from him. I would like to thank Dr. Azad Dehghan for giving me an opportunity to do an internship with of DeepCognito Ltd. during my PhD. I remember my beloved University of Dhaka and all the teachers at the Institute of Information Technology. The foundation of my knowledge was built during my bachelor's study at the University of Dhaka.

Finally, I want to thank Shikha Anirban, Khalid Hasan, Mojaharul Islam, Md. Mijanur Rahman, Emon Kumar Dey, Dr. Zhe Wang, Ryoma Ohira, Dr. Megha Khosla, and many others, for helping me directly or indirectly towards the successful completion of my PhD.

DEDICATION

To my beloved parents

Md Islam Uddin and Shamsunnahar

Contents

List of Figures

List of Tables

List of Abbreviations

ADE adverse drug event

ADR adverse drug reaction

AI artificial intelligence

BCQ binary causal question

BERT bidirectional encoder representations from transformers

BLSTM bidirectional long short-term memory

CEA cause-effect association

CNN convolutional neural network

COPA choice of plausible alternatives

CRF conditional random fields

EHR electronic health record

FFNN feed-forward neural networks

GRU gated recurrent unit

LSTM long short-term memory

MCNN multi-column convolutional neural networks

MHA multi-head attention

NATO the North Atlantic Treaty Organization

NB naive bayes

NE named entity

NLP natural language processing

NLU natural language understanding

PDTB penn discourse treebank

PMI point-wise mutual information

POS parts-of-speech

RHNB restricted hidden naive bayes

RNN recurrent neural network

SCAN shared causal attention network

SVM support vector machine

LIST OF PUBLICATIONS

Journal Articles

- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, Ryoma Ohira, and Zhe Wang, "SCAN: A Shared Causal Attention Network for Adverse Drug Reactions Detection in Tweets.," Neurocomputing, Volume 479, Pages 60-74, 2022.
- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, A. S. M. Kayes, and Paul A. Watters. "A deep learning model for mining and detecting causally related events in tweets." Concurrency and Computation: Practice and Experience, Volume 34, Issue 2, 2022.

Submitted Journal Article

- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, "Answering Binary Causal Questions Using Role-oriented Concept Embedding.", IEEE Transactions on Artificial Intelligence (*Submitted*)

Conference Articles

- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, Shikha Anirban, A. S. M. Kayes, and Paul Watters. "Answering binary causal questions: A transfer learning based approach." International Joint Conference on Neural Networks (IJCNN), Pages 1-9. IEEE, 2020.
- **Humayun Kayesh**, Md. Saiful Islam, and Junhu Wang. "Event causality detection in tweets by context word extension and neural networks." International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Pages 352-357, IEEE, 2019.

- **Humayun Kayesh**, Md. Saiful Islam, and Junhu Wang. "A causality driven approach to adverse drug reactions detection in tweets." International Conference on Advanced Data Mining and Applications (ADMA). Volume 11888, Pages 316-330, Springer, 2019. [*Best Paper Runner-up Award*]

Other Collaborative Articles

- Shikha Anirban, Junhu Wang, Md. Saiful Islam, **Humayun Kayesh**, Jianxin Li, Mao Lin Huang, "Compression techniques for 2-hop labeling for shortest distance queries.", World Wide Web, Volume 25, Issue 1, Pages 151–174, 2022
- Ryoma J. Ohira, Md. Saiful Islam, **Humayun Kayesh**,"Speedup vs. quality: Asynchronous and cluster-based distributed adaptive genetic algorithms for ordered problems." Parallel Computing, Volume 103, Pages 102755, 2021.
- Atish Kumar Dipongkor, Md. Saiful Islam, **Humayun Kayesh**, Md. Shafaeat Hossain, Adnan Anwar, Khandaker Abir Rahman, Imran Razzak, "DAAB: Deep Authorship Attribution in Bengali.", International Joint Conference on Neural Networks (IJCNN), Pages 1-9, 2021.
- Ryoma J. Ohira, Md. Saiful Islam, **Humayun Kayesh**, S. M. Riazul Islam, "MSGM: A Markov Model Based Similarity Guide Matrix for Optimising Ordered Problems by Balanced-Evolution Genetic Algorithms.", IEEE Access, Volume 8, Pages 210286-210300, 2020.
- A. S. M. Kayes, Md. Saiful Islam, Paul A. Watters, Alex Ng, **Humayun Kayesh**, "Automated measurement of attitudes towards social distancing using social media: A COVID-19 case study." First Monday, Volume 25, Issue 11, 2020.

Chapter 1

Introduction

Digital technology has enabled us to generate a high volume of data every day. Together we created 2.5 Quintilian bytes of data per day in 2020 [5]. A major share of this huge amount of data is text. Despite the recent advancement of natural language processing (NLP) we have been able to utilize a small portion of the text data available online. Causality is a piece of useful information that is hidden in the texts shared on different platforms online. Causal discovery denotes the automatic detection of causal relationships between entities. However, due to the huge size of the data, it is almost impossible for us to process this data manually and discover causality. Therefore, an automatic approach is necessary to extract and refine causality information from texts. Causal knowledge can be applied to almost every aspect of our life to improve our planning and assist in better decision-making [6].

1.1 Application Scenarios

Causality information has a wide range of applications such as prescriptive analysis, question answering, anomaly detection, and what-if analysis. Assume, we have a causality detection and analysis system as illustrated in Fig. 1.1. We exemplify a few practical applications of the system in the following scenarios:

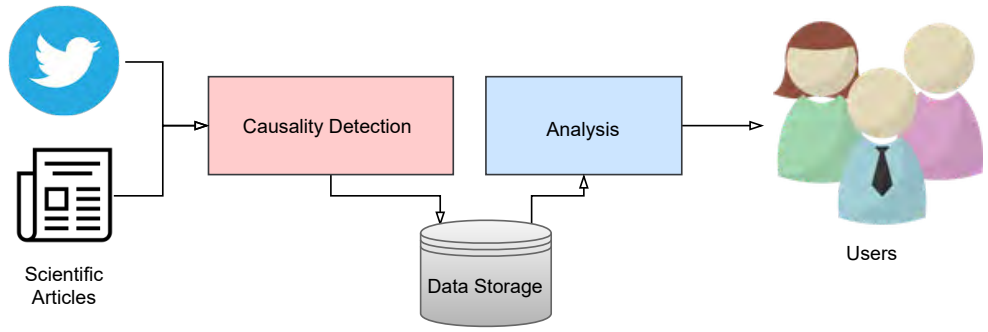


Fig. 1.1 A causality detection and analysis system

- **Scenario#1**: Consider a restaurant chain owner who has access to a causality detection and analysis system. Customers of the restaurant often use hashtags or mention the restaurant's social media handle to post any reviews, feedback, or comments on social media. The system can automatically collect such social media posts and extract causality information and analyze it. After running the system for some time, the system notifies the owner of the restaurant with some actionable insights. For example, if a customer posts the following Tweet, "Having upset stomach after eating at X restaurant", the system can detect the causal relationship between *upset stomach* and *eating at X restaurant* and notify the owner. This information should help the restaurant owner to take immediate actions to improve the hygiene of the restaurant. Another use-case of the system is action-impact analysis. The owner can check the possible effects of actions even before taking that action. Suppose, the owner wants to raise the price of a menu. The system can display the possible effects of price rise on customers based on the past data. This could help the restaurant owner to make better decisions.

- **Scenario#2**: Suppose, a pharmaceutical company produces and markets a Covid-19 vaccine and it wants to track the adverse effects of the vaccine on mass people. Vaccines pass through a series of trials in the production phase and a set of known side effects are identified before marketing. Such side effects are known as adverse drug reactions (ADRs). Lab tests and trials are performed in a very controlled environment and some side effects may not be identified until it is used by the mass people. Also, not everyone is interested in spending time to notify the company about the side effects. Instead, many patients use social media to share

their experiences of taking vaccines with others. Assume, a manager at the pharmaceutical company decides to use social media to discover unknown side effects of the vaccine. The manager has access to the causality detection and analysis system that can detect the causal relationships between the vaccine and the side effects mentioned in the posts on social media. For example, if a patient posts a Tweet, "Covid jab gave me headache #vaccine_name", the system detects that *headache* has been mentioned as a side effect of the vaccine. This should help the manager to identify new and unknown side effects of the vaccine.

- **Scenario#3** Finally, consider a health-related question answering portal that aims to provide answers to the questions on Covid-19. During a pandemic, providing the right information is crucial. Therefore, a portal that can answer basic questions about the causes and symptoms of various diseases could help to spread correct information. However, it is challenging to prepare a large collection of questions and answers. In this scenario, an automatic causal question answering system could be effective. A question-answering system requires a reliable source of information. The causality detection and analysis system can be used to discover causality in publicly available scientific papers. The system builds a causality knowledge base that contains the causes and symptoms of various diseases from the published articles which guarantees the quality of the automatically extracted information. The system with such a knowledge base should be able to answer questions like "Could Covid-19 cause lung failure?" or "Could smoking make Covid-19 infection worse?". Answering such causal questions with the latest information extracted from scientific articles could assist in providing general users with the correct information.

1.2 Problem Statement

In this section, we define the task we have undertaken in this thesis. Our goal is to discover causality in text data using deep learning techniques. The task includes extracting candidate events, concepts, or entities at first and then detecting causal

relationships between pairs of events, concepts, and entities. In this thesis, we focus on detecting causality based on only the available contextual information and commonsense knowledge. Fact-checking and verification of detected causality information are out of the scope of this thesis. We have formalized the research problem addressed in this thesis as below:

Definition 1.1. Given two concepts, events, or entities $\{X,\ Y\}$ in a context, automatically determine whether there is a causal relationship between X and Y.

In the above definition, if X influences or contributes to occurring Y, then X is the cause of Y, and Y is the effect of X. Also, $\{X,\ Y\}$ is considered to be in a causal relationship. Causality is a directed relationship and it is represented as $X \rightarrow Y$, if X is the cause and Y is the corresponding effect. In the following example Tweet, "Heavy rain caused a traffic jam in Gold Coast", the user mentioned two events. The first event is *heavy rain* and the second event is *traffic jam*. According to the contextual information in this Tweet, *heavy rain* and *traffic jam* are causally related, where *heavy rain* is the cause event and *traffic jam* is the effect event.

In this thesis, we aim to discover causality in text in three different contexts: (i) detecting causally related event pairs, (ii) detecting ADRs using causality, (iii) answering binary causal questions.

1.3 Challenges

NLP is a challenging sub-task of computer science and artificial intelligence (AI). Natural language is constantly evolving and unstructured in nature, that is why causal discovery in texts is a challenging task. We have discussed the challenges below in detail and referred to the section of this thesis in which we have addressed the challenges.

- **Detecting Causally Related Events in Tweets:** Social media short texts such as tweets are often written in a highly informal manner. Grammatical errors

are more frequent compared to the more formal writings such as scientific papers or newspaper. Causal discovery in such an unstructured dataset is a challenging task. The first challenge in causally related event detection is to detect the candidate events and the next challenge is to detect the causal relationship between the candidate events. The most trivial approach is to apply a set of linguistic rules to detect causal relationships [7–9]. However, due to the informal nature of Tweets, this approach is less effective (Section 3.4). Also, Tweets are short and often enough contextual features are not present to determine the causal relationship. We address the event representation challenge by proposing a sequence aware event representation technique (Section 3.3.2). We addressed the lack of contextual information using a context word extension technique (Section 3.3.4).

- **Detecting Adverse Drug Reactions in Tweets:** Patients often share their experiences of medication on Twitter. These tweets may contain ADRs and the associated drug names. Detecting ADRs in Tweets is challenging as Tweets are free text and informal. The same ADR can be written in various forms. Another challenge is to differentiate between ADRs and indications. The indication is a health condition for which patients take drugs. As both ADRs and indications often health conditions, it is challenging to differentiate between them. We have addressed the challenges mentioned above by proposing a causality-driven approach to detect ADR words in Tweets (Section 4.2.3). Unlike the event causality detection task, every word in a Tweet except the drug name could be an ADR. So, it is challenging to extract distinguishable contextual features from each word for ADR detection. We have addressed this challenge by proposing a novel causal feature extraction technique (Section 4.2.3.3).

- **Answering Binary Causal Questions:** A binary causal question contains a candidate cause concept and an effect concept. The question asks whether there is a causal relationship between the candidate cause and effect or not. The detection of causal relationships between two concepts requires rich background knowledge. Training a supervised model for causality detection requires large training data.

High-quality and large training datasets are often unavailable, expensive, or time-consuming to prepare. Even a model trained on large data may show poor performance on unknown data. We have addressed the problems by proposing a technique to prepare a training dataset using an automatically extracted publicly available dataset of causal pairs. However, the automatically extracted dataset is prone to noisy and incorrect causal pairs. The model trained on this dataset only suffers from low accuracy and imbalanced precision and recall. To address this problem we propose to use a publicly available causal concept bank that was extracted automatically but the quality was verified by a human. In this approach, finding the correct concept features from the concept bank is challenging as a concept may be written in various forms. We have addressed this challenge by using a semantic concepts similarity technique.

1.4 Contributions of This Thesis

The key contributions of this thesis are outlined below:

- **Detecting Causally Related Events in Tweets Using Deep Learning:** Our first contribution is a sequence aware event representation technique that is used to extract candidate causal events from Tweets. The structure of the proposed event representation technique is consists of an event keyword and a set of contextual words. The sequence-aware representation of events keeps the contextual words in their original order to preserve the positional features of the sequence. Our other key contribution is a novel context word extension technique that extends event contextual words using a background knowledge source. This approach addresses the lack of contextual information problems in event causality detection. We also propose a deep learning-based model that converts the candidate events into vectors and extracts features to detect causally related event pairs. We have discussed this in detail in Chapter 3.

  Detecting ADRs in Tweets using causality: One of our key contributions

is a deep neural networks model for ADR detection in Tweets using causality. We assume that there is a causal link between a drug name and an ADR word. The proposed model uses this assumption to extract causal features which are then shared with word features and parts-of-speech (POS) features. We have also proposed an efficient causal features extraction technique that splits a Tweet into segments and extracts features separately from each segment. We perform extensive experiments to show the effectiveness of your proposed techniques. We show that combining causal features with word and POS features improves ADR detection in text. These are presented in Chapter 4.

- **Answering Binary Causal Questions Using Concept Embedding:** We have proposed a deep learning-based framework to answer binary causal questions. We show that we can train a deep learning-based model to answer binary causal questions on carefully designed, automatically generated, small publicly available datasets. Our extensive experiments suggest that can achieve comparable or better performance compared to the state-of-the-art approach that requires a much larger dataset. Our other contribution is the application of the role-oriented causal embeddings of concepts to answer binary causal questions (BCQs). We show that using role-oriented concept embeddings can improve the accuracy of the answers to BCQs. This is discussed in detail in Chapter 5.

1.5 Structure of This Thesis

We have organized the rest of the thesis as follows:

- **Chapter 2 - Literature Review** discusses the background of causal discovery in text and provides an overview of the existing relevant literature. In this chapter, we have discussed various aspects of causality that serves as the foundation to follow the following chapters. We have also discussed existing approaches to event causality detection, ADR detection, and question answering. We have grouped similar approaches together and summarized them. We have

discussed the limitations of the existing approaches and compared against our proposed approach.

- **Chapter 3 - Detecting Causally Related Events** presents a deep learning-based approach to detect causally related events in tweets. In this chapter, we have discussed the challenges of causal discovery in social media short text such as informal writing and lack of contextual information. We have demonstrated that the lack of contextual information in causally related event detection can be addressed by using publicly available commonsense knowledge and a neural network-based model. The effectiveness of the proposed approaches is demonstrated by extensive experiments and discussion.

- **Chapter 4 - Causality for Adverse Drug Reactions detection** presents a causality-driven approach to ADR detection in Tweets. In this chapter, we have demonstrated that we use the cause-effect relationship between ADR and drugs to detect ADR words mentioned in Tweets. We have also presented a feature extraction technique that can be used to extract causal features for each word in a sentence. We have demonstrated that sharing causal features with word features and POS features improves the ADR words detection in Tweets. We have demonstrated the effectiveness of our proposed model by extensive experiments on three publicly available datasets.

  Chapter 5 - Answering Binary Causal Questions presents a deep learning-based framework to answer BCQs. In this chapter, we have demonstrated that we can train a deep learning model on automatically extracted datasets to address the challenge of acquiring a large dataset. We have also demonstrated an approach to combine causal concept embeddings with contextual features to improve the accuracy of answering of BCQs. We have also presented a technique to address the concept matching problem between train and test data in extracting causal features. We have demonstrated the effectiveness of our proposed approach by experimenting on five benchmark datasets.

- **Chapter 6 - Conclusions** summarizes the thesis contributions with concluding remarks and outlines future research directions in the context of causal discovery

in text. In this chapter, we have presented a summary of each chapter emphasizing the key findings. We have also discussed the possible extensions of the proposed approaches. Finally, for completeness, we finish the chapter by discussing a few areas in NLP that are relevant but out of the scope of this thesis.

Chapter 2

Literature Review

Recent advancement in natural language understanding (NLU) has allowed various fields including causal discovery to achieve significant progress. In this chapter, we will discuss the background of causality and causal discovery in text. We will also study the ongoing progress of causal discovery. We will provide a summary of various causality detection approaches and compare them against our proposed approaches.

2.1 Causality

Textual data is a source of human knowledge. Entity-relationship is such a form of knowledge that is often expressed in text. Causality is referred to a relationship between two entities where one entity causes another entity or one entity is a consequence of another entity. Such a relationship is often represented as "X causes Y" or "Y is caused by X" where X is the cause entity and Y is the effect entity. Depending on how causality is expressed, we can categorize the causal relationship into two categories: the explicit and implicit causality [10]. The explicit causality contains both cause and effect entities in a given text and the relationship between entities is clearly expressed. However, in implicit causality, the relationship is not clearly expressed, and often either cause or effect entity is hidden.

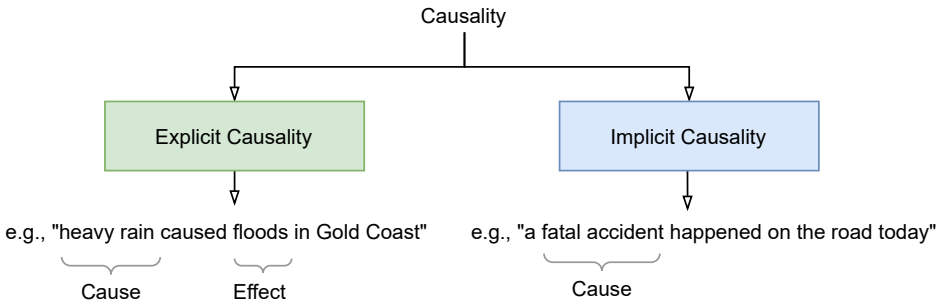


Fig. 2.1 Types of causality

Fig. 2.1 presents two examples of explicit and implicit causality. The first example sentence, "heavy rain caused floods in Gold Coast", contains an explicit causal relationship. In this example, *heavy rain* is the cause entity and *floods* is the effect entity. The relationship between *heavy rain* and *floods* is clearly expressed the sentence. However, the second sentence, "a fatal accident happened on the road today", contains an implicit causality. In this sentence, *fatal accident* is the cause but the effect *death* is not explicitly mentioned.

2.2 Representation of Causality

Two entities in a text may have various relationships and causality is one such relationship. Entities in causal relationships are often represented as events [10]. An event contains semantic and syntactic information about that entity. For example, in the following causal relationship, "X causes Y", X is denoted to be a cause event and Y is denoted to be an effect event. A cause or effect event contains is consists of two key components: keyword and context words. A keyword in the event contains the key information about the entity and the other contextual information is captured by the context words. For example, the following sentence, "Mike had a car accident in Gold Coast", *accident* is the word that contains the key information about the event. The other words, such as *Mike, car, Gold Coast*, contains additional information about the event. There the event could be represented by *accident(Mike, car, Gold Coast)*.

2.3 Causal Discovery in Text

Text is a commonly available data form and entity-relationship detection is a fundamental problem in NLP. Causal discovery in texts denotes the task of detecting the causal relationships between entities. The aim of the task is to extract causal information from textual data. Causal information has a wide range of applications, such as prescriptive analysis [1], scenario prediction [11, 12], question answering [13], and next actions prediction [14].

The causal discovery task requires a model that can understand the context and calculate the causal strength between entities [7]. The higher causal strength indicates a stronger causal link. The detection of causal strength between entities often requires background knowledge. Also, it is challenging to develop training datasets for causality detection due to the low frequency and ambiguous nature of causal relationships. Therefore, existing causal discovery approaches often use various sources of background knowledge, such as newspaper articles, social media posts, scientific articles, and electronic health records (EHRs). Most of the above sources are publicly available and easily accessible.

2.3.1 Rule-based Approaches

In this section, we discuss the causality detection approaches that applied linguistic rules and patterns. Among the rule-based approaches to causality detection, a few approaches used hand-crafted rules while others explored the automatic generation of rules from training data. Fig. 2.1 displays a summary of the rule-based approaches to causality detection.

Table 2.1 Rule-based approaches to causality detection

Approaches	**Aim**	**Methods / Tools**	**Supervision**	**Corpus**	**Drawbacks**
Marcu and Echihabi [15]	Detecting discourse relations	Hand-crafted linguistic rules, naive bayes (NB) classifiers	Unsupervised	41 Million Enligsh sentences, BLIPP [16]	A small set of rules used for caual relations extraction
Girgu [13]	Causal relationships detection for question anwering	Automatically generated lexico-syntactic patterns	Unsupervised	WordNet	Only extracts noun-noun relationship
Chang et al. [17]	Causality detection between event pairs	NB classifiers	Unsupervised	TREC [18], medical encyclopedia	Only uses lexical similarity to determine causal relationships
Blanco et al. [19]	Causal relations detection in open text	Hand-crafted linguistic rules, C4.5 decision trees [20]	Supervised	SemCor 2.1 [21]	A small dataset used for training and evaluation
Riaz and Girju [22]	Detecting causally related verbal events in texts	The causal association between verb-verb pairs calculated using point-wise mutual information (PMI) scores	Supervised	240K instances extracted from English Gigaword corpus	Only focused on verb-verb relations
Khan et al. [1]	Detecting causally related event objects in system logs	Association rule mining	Unsupervised	System error logs from ten machines	Not suitable for causality in text

continued ...

. . . continued

Approaches	Aim	Methods / Tools	Supervision	Corpus	Drawbacks
Luo et al. [7]	Detecting commonsense causality in short text	Linguistic rules and causal network	Unsupervised	1.6 billion web pages	The CasualNet contains single words only

One of the early approaches to entity relations detection in texts was a rule-based approach proposed by Marcu and Echihabi [15]. The unsupervised approach automatically generates a corpus from 41 million English sentences using a set of hand-crafted linguistic rules. A NB model is then used to classify the discourse relations. However, the generation of hand-crafted rules often requires expert knowledge of the dataset. Addressing that challenge, an automatic rule generation technique was proposed by Girgu [13]. The author proposed an automatic approach to generate lexico-semantic patterns to answer causal questions. Their approach only focuses on extracting noun-noun that conforms to the following pattern: $< NP_1\ verb\ NP_2 >$, where NP corresponds to a noun phrase. The author used the *CAUSE-TO* transitive relation in WordNet [23] to extract noun phrase pair and then a collection of texts was used to extract the lexico-syntactic patterns.

Causal relationships are often represented as events and Chang et al. [17] proposed an unsupervised technique to detect causal relationships between events. In contrast to [13], the authors used mutual information in event pairs to detect causal relationships rather than using causal verbs between nouns. The authors assumed that if two event pairs have lexical similarities and one of them is known to be causally related, then there is a high chance that the other pair is also causally related. A NB model that applies the lexical and cue phrase probabilities learned from raw texts was used to extract the causal relationships between event pairs.

Blanco et al. [19] proposed an approach that uses hand-crafted linguistic patterns to detect causality in open texts. First, the authors developed a set of linguistic rules and then trained a supervised classifier to detect the applicability of the rules for causality detection. This approach only focuses on the causal relationships between the verb phrases and clauses that are connected by a *relator*. For example, in "I *went* to there because *I was invited*", the verb phrase *went* is connected to the clause *I was invited* by the *because* relator. The authors trained a C4.5 decision tree [20] model to classify the causal patterns.

Like Blanco et al. [17], Riaz and Girju [22] proposed an event causality detection

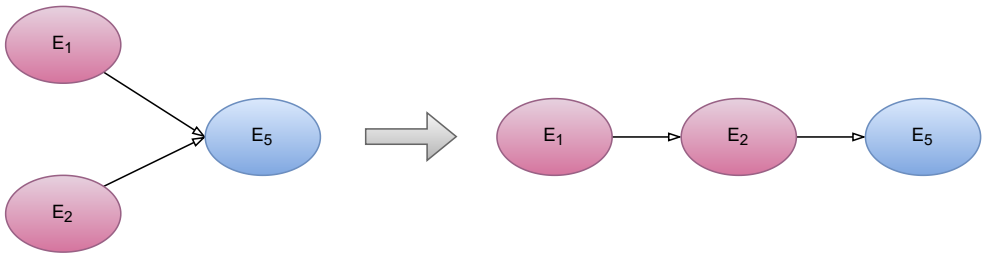


Fig. 2.2 Generating a causal chain from an association rule (adapted from [1])

approach. The proposed technique automatically extracts verb-verb event pairs from texts and uses that data to detect causality in event pairs. First, the authors prepared a corpus of 240k examples from the English Gigaword Corpus. The discourse marker *because* was used to extract the positive instances and *but* was used to extract negative instances when building the corpus. The PMI scores were calculated using the corpus to detect the causal likelihood of verb-verb event pairs. The authors found the lexical, syntactic, and semantic features to be useful in verbal event causality detection. Another event causality detection approach was proposed by Khan and Parkinson [1]. The authors applied an association rule mining technique [24] to automatically detect the casually related security events from system logs. The key contribution of this approach is generating a causal chain from causally related events. First, a set of strong association rules are generated from the frequently occurring events. Then the rules are combined together to generate a causal chain. Fig. 2.2 illustrates an example of causal chain generation between three events, E_1, E_2, and E_5. In the example, the chain $E_1 \rightarrow E_2 \rightarrow E_5$ is generated from the rule $\{E_1, E_2\} \rightarrow E_5$. The relative order between E_1, E_2 in the causal chain is determined by their order of starting time. In the example scenario, E_1 starts before E_1 in the system log. However, the key drawback of this approach is that the causal relationships between the events on the left-hand side are not verified when merging them into a causal chain.

Luo et al. [7] proposed a causality detection technique that extracts pairs of causal phrases using linguistic and rules and commonsense knowledge. The causal pairs were then used to develop a causal network. Each node in the causal network is a single word and edges contain the frequency of words pairs. The causal network was then used to calculate causal strength. However, the causal network only contains single words.

2.3.2 Machine Learning-based Approaches

In this section, we discuss the existing approaches to causality detection that used machine learning techniques, such as NB, support vector machine (SVM), and conditional random fields (CRF) [25]. Such approaches address different sub-tasks of causality detection such as commonsense causality detection, causally related events detection, and the causal relationship detection between drugs and ADRs. The majority of the existing machine learning-based approaches used supervised learning while a few approaches explored the semi-supervised approach. A wide range of feature extraction were key steps in the existing approaches. The approaches used lexical, syntactic, semantic, and temporal features to detect causality. A summary of the machine learning-based approaches to causality detection is presented in Table 2.2.

Table 2.2 Machine Learning-based approaches to causality detection

Approaches	**Aim**	**Methods / Tools**	**Supervision**	**Corpus**	**Drawbacks**
Do et al. [26]	Causality detection between noun and verb events	Distributed similariity and constrained conditional models [27]	Semi-supervised	penn discourse treebank (PDTB) [28]	Only focus on nouns and verbs
Roemmele et al. [29]	Proposing a framework for evaluating commonsense causality detection	PMI	Supervised	One million personal stories from Internet weblogs [30]	Limited usage of background knowledge
Rink and Harabagiu [31]	Semantic relation classification for SemEval-2010 Task 8	SVM trained on lexical and semantic features	Supervised	WordNet, VerbNet, NomLex-Plus, and Google N-gram [32]	Requires extensive features engineering
Pal et al. [33]	Semantic relation classification for SemEval-2010 Task 8	CRF	Supervised	WordNet	Nominal pairs in sentences are required to be marked
Zhao et al. [34]	Extraction of causality information from texts	Used the categories of causal connectives as a feature to train a restricted hidden naive bayes (RHNB) model	Supervised	2682 sentences with 50% having causal relation	Single-pair causality detection in sentences
Radinsky et al. [11]	Predicting the future effect events for a given cause event	Causality patterns to prepare a causal graph, clustering	Supervised	LinkedData that contains 20 billion event relations, and WordNet	Requires a large training data to train

continued ...

... continued

Approaches	Aim	Methods / Tools	Supervision	Corpus	Drawbacks
Riaz and Girju [35]	Detetion of causality between verb-noun pairs using lexical, semantic, structural features	NB	Supervised	FrameNet and WordNet	Loss of information
Yang and Mao [36]	Extraction of causal relations between clauses	An SVM model trained on positional, syntactic and semantic features	Supervised	WordNet, VerbNet, and FrameNet	Overfitting problem
Hidey and McKeown [37]	Use linguistic markers to detet causality in texts	Linear SVM	Semi-supervised	Wikipedia articles	Only addresses a sub-task of the causality detection problem
Bethard et al. [38]	Joint detection of both temporal and causal relations between events	Syntactic and semantic features, SVM	Supervised	WordNet, Google N-gram	A Small datasets used for evaluation
Rink et al. [39]	Event causality detection in texts	Graph patterns extraction, SVM	Supervised	1000 English sentences	A Small datasets used for evaluation
Mirza [40]	Proposing a framework to annotate the temporal and causal relationships	SVM trained on temporal and morpho-syntactic features	Supervised	TimeBank [41] and AQUAINT [42]	
Mirza and Tonelli [43]	Extraction of temporal and causal relations between events	Syntactic rules and SVM	Supervised	Causal-TimeBank [1]	Linguistic rules require expert knowledge

continued ...

[1] http://hlt-nlp.fbk.eu/technologies/causal-timebank

. . . continued

Approaches	Aim	Methods / Tools	Supervision	Corpus	Drawbacks
Ning et al. [44, 45]	Detection of temporal and causal relations in texts	Linguistic rules, constrained conditional models [46]	Supervised	TimeBank-Dense [47], EventCausality [26], and Causal-TimeBank [48]	Limited support for event causality detection

Do et al. [26] proposed a semi-supervised machine learning-based approach to detect causal relationships between events. This is one of the early approaches to use discourse distributional similarity and discourse connectives for causality detection. Firstly, the authors extract events from a given text document and then calculate the cause-effect association between events. They trained a constrained conditional model [27] using the co-occurrence scores between events. The co-occurrence scores were learned automatically from unlabeled data. The model also uses the discourse relations between events which were extracted from PDTB [28] corpus. In the following example "He was happy because he passed the exam", the events *happy* and *passed* are connected by the connective *because*. The model learns from the corpus that there is a strong causal relationship between *happy* and *passed* as they are connected by connective *because*. However, the model only focuses on the events with the noun and verb word pairs in a text document.

It is challenging to evaluate commonsense causality detection approaches due to a lack of a proper evaluation framework. Roemmele et al. [29] proposed a framework to evaluate open-domain commonsense causality detection approaches. The authors also prepared a dataset of one thousand causal inference questions named choice of plausible alternatives (COPA) for evaluating commonsense causality reasoning. Each question contains a premise and two possible alternatives. The authors proposed a benchmark three benchmark approaches for causality detection that use PMI to detect the causal relationships between the questions and the answer choices. Eq. 2.1, which is adapted from [29], shows the calculation of PMI score between a premise, p and two alternatives a_1 and a_2.

$$PMI = argmax_{a\epsilon\{a_1,a_2\}} \frac{hits(p+a)}{hits(a)} \tag{2.1}$$

Background knowledge is important for causal reasoning. One of the early approaches that extract rich background knowledge to detect causality in texts was proposed by Rink and Harabagiu [31]. This approach detects semantic relationships between nominals, which is task 8 in the SemEval-2010 challenge [49]. The model

Table 2.3 Different types of expressions of the same cause-effect relation

#	**Examples**	**Causal Relationships**
1	Civil war is one of the main *causes* of migration	Civil war $\xrightarrow{\text{cause}}$ migration
2	Civil war *increases* migration	Civil war $\xrightarrow{\text{increase}}$ migration

extracts lexical and semantic features from lexical databases such as WordNet [2], VerbNet [3], NomLex-Plus[4], and Google N-gram [5] data. However, this approach requires extensive feature engineering. The same task was attempted by Pal et al. [33] and the authors applied a CRF to detect semantic relations between nominal pairs. The model was trained on semantic features, e.g. WordNet synsets and named entity (NE), syntactic features, e.g. dependency relations of words and lexical features, e.g. list of verbs associated with the target semantic relations. The authors found that the NE features play an important role in semantic relations detection.

The same causal relationship can be expressed in different ways using different causal connectives. Table 2.3 displays two such examples. In the first example, "Civil war is one of the main *causes* of migration", two events, *civil war* and *migration*, is connected by the connective, *cause*. However, in the second example, "Civil war *increases* migration", the same pairs of events are expressed by the connective, *increase*. Zhao et al. [34] proposed an approach that considers the usage of causal connectives as an important feature to train their causally related events detection model. The authors used dependency parsing to categorize the connectives and use this feature to train a RHNB model. The model was trained on a corpus of 2682 sentences with 50% of them having causal relation and evaluated on the SemEval-2010 task 8 dataset.

Radinsky et al. [11] proposed an automatic approach to predict future events. The authors prepared a large training corpus from newspaper titles using a set of causality patterns. The corpus was then used to build a knowledge graph that contains an object in each node and a label in each edge, e.g. *capitalOf*. In this approach, an abject is a

[2]http://wordnet.princeton.edu/
[3]http://verbs.colorado.edu/ mpalmer/projects/verbnet.html
[4]http://nlp.cs.nyu.edu/meyers/NomBank.html
[5]https://catalog.ldc.upenn.edu/LDC2009T25

constituent part of an event. Based on the knowledge graph, the authors automatically generated linguistic rules for event causality prediction. A clustering technique that minimizes the variance of causes and effects among events was applied to group similar events together. A supervised model was then trained to learn linguistic rules for future event prediction. However, the model requires large training data to generalize well.

Another NB-based approach was proposed by Riaz and Girju [35] that detects causality between verb-noun pairs using lexical, semantic, structural features. The supervised approach extracts features from linguistic knowledge sources, such as WordNet and FrameNet. The authors used the lemmatized version of verbs and nouns as the lexical features. The binary encoding of 9 noun hierarchies of WordNet was used as the semantic features. The authors also extracted the structural features extracted from the position of subjects and objects in noun phrases. In contrast to [35], Yang and Mao [36] proposed an approach that detects causality between any clauses rather than limiting the scope to the noun or verb phrases only. The authors trained an SVM model using various positional, syntactic, and semantic features extracted from WordNet, VerbNet, and FrameNet. However, this model is prone to the overfitting problem. Another approach that uses linguistic knowledge sources to detect causality in texts was proposed by Hidey and McKeown [37]. Following [22], the authors used a semi-supervised technique to automatically learn the linguistic markers that express causality from the Simple and English Wikipedia articles.

Causal relationships between events are sometimes associated with temporal relationships. Bethard et al. [38] proposed one of the early approaches to jointly detect both temporal and causal relations between events. The supervised approach trains a NB-based model using both syntactic and semantic features extracted from the WordNet and Google N-gram corpora. The syntactic features include event words, lemmatized version of words, POS tags, and word dependency relations. The semantic feature set consists of event root synsets and lexicographical file names extracted from WordNet. To solve the same problem, Rink et al. [39] proposed an approach that extracts graph patterns to train a SVM model to detect causal and temporal relationships. However, these models were evaluated on a small dataset of 1000 English

sentences.

Mirza [40] proposed an annotation framework to detect both temporal and causal relations in text. The author employs temporal constraint on the causality relation following the TimeML corpus annotation framework [50]. Later, Mirza and Tonelli [43] proposed a hybrid approach that combines rule-based and machine learning-based techniques to detect temporal and causal relationships. The linguistic rules, which were used to extract temporal relations, were generated based on the entities in the inputs and their grammatical structures. The machine learning part of the model, which is responsible for the causal relationship classification, consists of a linear SVM model. The model was trained on the Causal-TimeBank dataset [48]. However, the generation of hand-crafted rules requires expert knowledge of the data. Ning et al. [44, 45] proposed a similar approach that combines linguistic rules and machine learning techniques for joint detection of temporal and causal relationships. Unlike Mirza [40], the authors used constrained conditional models in the causal relation component instead of using SVM. The model was trained using the TimeBank-Dense [47], EventCausality [26], and Causal-TimeBank [48] datasets.

2.3.3 Deep Learning-based Approaches

This part of the chapter reviews the deep learning-based technique to discover causality. The early approaches that explored causality applied feed-forward neural networks (FFNN) and recurrent neural network (RNN) [51]-base techniques. The later approaches applied different variations of convolutional neural network (CNN) and hybrid approaches that extract features using linguistic rules and then train a deep learning model on them. The advancement of transformers in NLP allowed recent approaches to apply transfer learning-based techniques to causality detection. Table 2.4 summarizes a the deep learning techniques to causality detection.

Table 2.4 Deep Learning-based approaches to causality detection

Approaches	**Aim**	**Methods / Tools**	**Supervision**	**Corpus**	**Drawbacks**
Rosario and Hearst [52]	Semantic relations detection between two-word noun compunds	A FFNN with one hidden layer	Supervised	2245 noun components extracted from biomedical articles	A small dataset was used for training and evaluation
Ponti and Korhonen [10]	Causally related events detection in text	Positional and semantic features, FFNN	Supervised	PDTB [28] and CSTNews corpus [53]	Relies on the word positional features
Roemmele and Gordon [12]	Modeling causality between events in stories	Encoder-decoder models with FFNN and RNN	Supervised	COPA [29], ROCStories [54]	Not suitable for long text sequences
Dasgupta et al. [55]	Extracting causal relationships in text	linguistic features and a bidirectional long short-term memory (BLSTM) [56]	Supervised	SemEval, adverse drug effect, and the BBC news articles dataset, analyst reports, and the recall dataset [6]	Treats every word equally in the cause/effect phrases
Nauta et al. [57]	Causal relationships detection in time series data	Attention mechanism with CNN	Supervised	Simulated datasets in the financial market and functional magnetic resonance imaging	Experiments were run on the simulated datasets only
Li and Mao [58]	Causal relations extractions in texts	Linguistic, lexical and semantic features; CNN	Supervised	WordNet and FrameNet	Loss of information
Bollegala et al. [59]	ADR detection in tweets	Lexical patterns and CNN	Supervised	94,890 Tweets	Used only the syntactic features

continued ...

[6] https://www.edmunds.com/recalls/

. . . continued

Approaches	Aim	Methods / Tools	Supervision	Corpus	Drawbacks
Kruengkrai [60]	Detecting event causality in text	multi-column convolutional neural networks (MCNN) [61]	Supervised	4 billion web articles	Requires a large datasets to train
Oh et al. [62]	Causality-related why-questions answering	MCNN	Supervised	35 million sentences extracted from the Japanese Wikipedia	Evauated on the Japanese language dataset
Kadowaki et al. [63]	Event causality detection in texts	Ensembles and bidirectional encoder representations from transformers (BERT) [64]	Supervised	Wikipedia and web articles	Requires expert kowledge to train the model
Khetan et al. [65]	Detecting causally related events in texts	BERT	Supervised	SemEval, adverse drug effect [66]	Only focuses on the intra-sentence causality detection
Li et al. [14]	Conditional generation of cause and effect texts	Transformer-based [67] conditional generation models	Semi-supervised	Common crawl corpus [68] and a variation of CauseNet [7]	Requires a large training data

Rosario and Hearst [52] was one of the early approaches that explored neural network-based techniques to detect semantic similarity in text. The authors proposed a simple FFNN with only one hidden layer to detect semantic relations between two-word noun compounds. However, the model was trained and evaluated on a small dataset of 2245 noun components extracted from biomedical articles. Ponti and Korhonen [10] proposed another approach that uses FFNN to detect causally related events in texts. While the model structure is similar to [52], the authors focused on enhancing the feature set. The model was trained on both positional and semantic features. Dependency parsing was applied to the candidate phrases to find the event keywords and participants. The event words are then lemmatized and vectorized to be used as semantic features. The positional features include the distance between two events words and the other words in event pairs. The model was trained on the PDTB [28] and CSTNews [53] datasets.

Roemmele and Gordon [12] proposed an encoder-decoder model to detect causality between events in stories. The authors proposed two variations of the model, one version of the model uses a FFNN and the other version uses a RNN. The model were trained on the COPA [29] and ROCStories [54] datasets. However, such sequence to sequence approaches often suffer for long input sequences as RNN may forget contextual information for longer sequences. Dasgupta et al. [55] address this challenge by using a BLSTM [56] to extract causal relations in text. The authors used linguistic features to train the model, which was evaluated on the SemEval, adverse drug effect, BBC news articles, analyst reports, and the recall dataset[7].

A CNN was initially applied to computer vision and image processing but later it was found to be effective in a wide range of other tasks including causal discovery. Nauta et al. [57] proposed attention-based CNN model to discover causality in time series data. The model is consists of n units of attention-based CNNs where each unit is responsible for predicting a different target series. The authors also discovered temporal relationships in observational data and constructed a causal graph. The model was evaluated on two simulated datasets of financial market and functional magnetic resonance imaging. Another CNN-based approach to causality extraction in texts was proposed by Li and

[7]https://www.edmunds.com/recalls/

Mao [58]. The authors extracted linguistic, lexical, and semantic features from WordNet and FrameNet to train the model. Similarly, Bollegala et al. [59] proposed a CNN-based approach to detect cause-effect relationships between drugs and ADRs in Tweets. However, the authors only focus on syntactic features, such as lexical patterns, to train the model. A dataset of 94,890 Tweets was used to train the supervised learning model.

Kruengkrai et al. [60] proposed an event causality detection approach that uses a MCNN [61] to detect causal relationships between events in texts. However, the model requires a large dataset to generalize well. The authors extracted cause-effect event pairs and detected systematic patterns from a dataset of 4 billion web articles to train the model. The authors argued that linguistic patterns play a significant role in event causality detection. Another MCNN-based approach was proposed by Oh et al. [62] for causality-related why-questions answering. Like [60], this approach also relies on large background knowledge. The model was trained on 35 million sentences extracted from the Japanese Wikipedia.

Recent approaches to causality detection explored transfer learning-based techniques. Kadowaki et al. [63] proposed a BERT-based approach to detect causality between events in texts. The authors used the annotation policy of human annotators as the features to train the model. The authors argued that pretraining a model on a large dataset of causal events improves the performance of a causality detection model. However, this approach requires expert knowledge of the dataset to train the model. Another BERT-based approach was proposed by Khetan et al. [65] that detects causally related events in the same sentence. The authors explored a language modeling technique that combines both sentence and event contexts to detect event-event causal associations. The model was evaluated on the SemEval and adverse drug event (ADE) [66] datasets. Li et al. [14] proposed a transformer-based [67] generative approach that uses conditional generation models. The authors focused on generating possible causes and effects from given text input. The model uses large corpora of background knowledge such as common crawl corpus [68] and a variation of CauseNet [7, 69].

2.4 Our Approach vs Existing Approaches

2.4.1 Detecting Causally Related Events

Our literature review found two broad categories of casualty detection approaches: non-event causality detection and event causality detection. The first group does not extract events before detecting causality whereas the second group extracts candidate events first, and then detect causal relationships between event pairs.

The early non-event causality detection approaches focused on word-to-word causality relationship detection [13,22] and applied linguistic rules and patterns to detect causality [7,15,19]. Many approaches [29,31,33,35,36] used machine learning-based techniques with rich background knowledge extracted from lexical databases such WordNet, VerbNet, and Google N-grams. Transformers and transfer learning-based techniques were also applied to detect non-event causality in text [14]. However, the approaches that extract events before detecting causal relationships are more relevant to our approach.

Many event causality detection approaches extracted event words and attributes and then applied word association techniques to detect causality between events [1, 22, 26]. Lexico syntactic rules and patterns were also found to be effective in event causality effective. Supervised machine learning-based approaches such SVM and NB were also proposed [17, 39]. Many approaches extracted temporal relationships between events [43–45, 48]. However, temporal information of events is often missing in short texts such as Tweets.

Many recent approaches to event causality detection applied neural networks-based techniques [60,70]. Among the event causality detection approaches, the neural network-based approach proposed by Ponti et al. [10] is the closest to our proposed approach. The authors used the word-to-word distance between event keywords and attributes as features to train their neural network model. However, the word distance features often become noisy when extracted from social media short texts such as Tweets. Also, often, limited contextual information is available in Tweets. Therefore, in Chapter 3, we proposed a

novel contextual word extension technique that extends context features using background commonsense knowledge. There are other various approaches [71–74] in the literature that were proposed to detect causality. A detailed literature survey on deep learning and machine learning-based techniques to causality detection was published by Ali et al. [75].

2.4.2 Causality for Adverse Drug Reactions Detection

In this section, we discuss the existing approaches to ADR detection that are relevant to our proposed approach. Early approaches to ADR detection apply association rule-mining approaches. Yang et al. [76] proposed a technique that applies association rule mining to detect ADRs in Tweets. The authors calculated proportional reporting ratios [77, 78] to find the strength of co-occurrence between drug names and ADRs words. Qin et al. [79] proposed an association clustering-based approach to detect multi-drug ADRs. The authors automatically generated association rules and applied a pruning technique to discard less important rules. Association rule mining techniques were also applied to detect and analyze ADRs in online texts such as health-related social media platforms [80, 81]. The above-mentioned techniques only consider the association between drugs and ADRs, but semantic relationships between drugs and ADR words have not been explored much.

Many existing approaches apply supervised learning techniques including both machine learning and neural network models. Bollegala et al. [59] proposed an approach that applies SVM [82] to detect whether a Tweet contains an ADR or not. The authors extracted lexical features using the skip-gram technique and used them to train the SVM model. Huynh et al. [83] proposed an approach that uses word features to train a neural network-based model for ADR detection.

Wu et al. [84] proposed an approach that detects both drug names ADRs in Tweets. The authors extracted both character and word features and applied and an multi-head attention (MHA) model for ADR detection. Chu et al. [85] proposed a model to detect adverse medical events using an attention based technique. The authors extract semantic

features to train their model. The combination of both contextual and semantic features were also found to be effective in neural network-based models [86, 87]. Zhang et al. [87] applied a CNN and a capsule network to extract contextual and semantic features, respectively. A medical relations classification approach was proposed by Luo [88]. The model combines a RNN [51, 89] and a long short-term memory (LSTM) [90] to detect ADRs in texts.

Many existing approaches approached the ADR detection task as a sequence labeling problem. The approaches predicted a label per word rather than a label per sentence. Song et al. [91] proposed an approach that applies CRF [25] to detect both ADRs and indications [92]. The model uses n-grams and POS tags as the features. Similar techniques were found to be effective by many other existing approaches [93–95]. Recent approaches also explored neural network-based techniques to detect ADR words in texts. Chowdhury et al. [96] proposed an approach to classify whether a text contains any ADR or not. The RNN-based model also performs sequence labelling for ADRs and indications. Cocos et al. [4] proposed an approach that uses a BLSTM model to detect ADRs in Tweets. Florez et al. [97] proposed a similar approach. The authors used BLSTM to detect ADR in medical texts. Another approach was proposed by the same authors that uses an improved feature extraction strategy [98]. A combination of CRF and BLSTM was also found to be effective in ADR phrase detection by Tutubalina and Nikolenko [99]. The authors proposed to combine both character and word embeddings to train their model in a dataset of review comments. A similar feature extraction strategy was applied by Ding et al. [100]. The authors extracted character and word embeddings from health-related texts trained their model to detect ADR. El-allaly [101] proposed an approach that jointly detects the boundary and type of ADEs mentioned in texts. Recent approaches also explored the application of BERT and graph neural networks in ADR detection [102–105].

Many existing approaches detect ADRs in EHRs instead of publicly available texts. A recent approach proposed by Wei et al. [106] found neural network-based techniques to be effective in this task. Panday et al. [107] proposed an approach that uses a BLSTM model and the model was trained using the word embeddings that were learned from clinical

texts. Another BLSTM-based approach was proposed by Jagannatha and Yu [108] that detects medical events in EHR. The authors combine a BLSTM model with a gated recurrent units (GRUs) [109] model. The combination of BLSTM and CRF was also found to be effective in ADE detection by Wunnava et al. [110]. However, EHRs often contains sensitive information of patients and is not publicly available. Also, the above-mentioned approaches do not consider the cause-effect relationship between drugs and ADR. Therefore, in Chapter 4, we have proposed a causality-driven approach to detect ADR words in Tweets.

2.4.3 Answering Binary Causal Questions

The answering BCQ task spans both causality detection and question-answering domains. For discussion on the causality detection approaches, we refer to Section 2.3. The existing approaches to causality detection discussed in Section 2.3 do not directly address the BCQ task. In the literature, we found only two significant works that addressed the answering BCQ task. The first one is a transfer learning-based approach proposed by Hassanzadeh et al. [111]. The authors used 17 million causal sentences to train their model. The sentences are first converted into vectors using BERT and top-k causal sentences are extracted using a k-Nearest Neighbour search. The authors calculated the average cosine similarity of top-k sentences with the sentences: "X may cause Y" and "Y may cause X", where X and Y are two candidate causal concepts. Answers to the binary causal questions are determined by the above-calculated scores and two thresholds. However, the model requires a large dataset for training, and expert knowledge on test datasets is required to set the threshold values. The second approach on answering BCQ is proposed by Kayesh et al. [112]. The authors proposed a transfer learning-based approach that requires a comparatively smaller training dataset. However, both of the approaches achieve low accuracy and precision scores on test datasets. Therefore, in Chapter 5, we proposed to use a role-oriented concept embedding and a semantic concept similarity technique to address this issue.

STATEMENT OF CONTRIBUTION TO CO-AUTHORED PUBLISHED PAPER

Chapter 3 includes a co-authored journal paper, which has been published in Concurrency and Computation: Practice and Experience in 2020. The bibliographic details of the co-authored paper, including all authors, are:

- **Humayun Kayesh**, Md Saiful Islam, Junhu Wang, A. S. M. Kayes, and Paul A. Watters. "A deep learning model for mining and detecting causally related events in tweets." Concurrency and Computation: Practice and Experience (2020), Volume 34, Issue 2.

My contribution to the paper involved: proposal of deep causal event detection technique, implementation, experiments, writing and editing manuscript.

(Signed)  (Date) June 10, 2022

Humayun Kayesh

(Countersigned)  (Date) June 10, 2022

Corresponding author of paper: Md. Saiful Islam

(Countersigned)  (Date) June 10, 2022

Supervisor: Junhu Wang

Chapter 3

Detecting Causally Related Events

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 chapter, 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.

3.1 Introduction

The "smart city" concept has drawn researchers' attention from various fields. 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 kind 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 e.g. 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.

Motivating Example. Fig. 3.1 illustrates a hypothetical example of event causality detection in tweets and its use in prescriptive analysis. The example tweet in Fig. 3.1: "A disruption in bus service in Gold Coast due to lack of communication between translink and event organizers" contains two causally related events. 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, built 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.

Applications. Other application areas of causality detection in social media include political events analysis [113], income analysis [114], career path prediction [115], adverse drug reaction (ADR) detection [74], and automatic question answering [112]. For instance, post-marketing 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

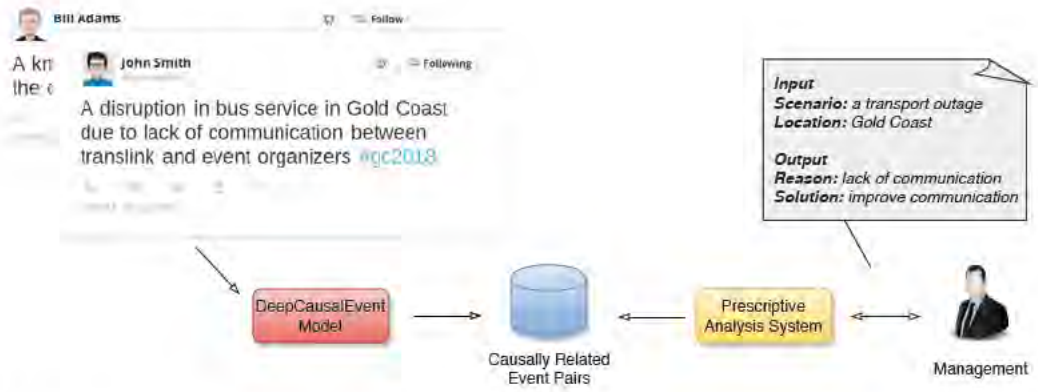


Fig. 3.1 Application of automatic event causality detection in predictive event analysis (adapted from Kayesh et al. [2])

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 [4].

Challenges. Event causality detection is a challenging natural language processing (NLP) task and it is an evolving area of research [7, 19, 22, 26, 35, 40, 60, 61]. In fact, causality detection in short text such as tweets is more challenging than relatively more formal text such as news articles [116]. The existing approaches that apply hand-crafted rules [7–9] are less effective in tweets (see section 3.4) 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. [10] is not much effective on detecting event causality in tweets. To deal with this issue, in this chapter we propose to apply a context word extension technique that can add additional contextual words based on background knowledge.

Contributions. In this chapter, we propose an improved version of our preliminary work [2] (technical report [117]) 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 multi-head attention mechanism. Finally, we apply two parallel two-dimensional convolutional neural networks (CNNs) followed by two-dimensional 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 chapter 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.

Organization. We present the remaining sections of this chapter in the following order: Section 3.2 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 3.3; the experimental results and discussions are illustrated in Section 3.4, and the conclusion remarks are presented in Section 3.5.

Table 3.1 Representation of Events (adapted from Kayesh et al. [2])

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)

3.2 Problem Formulation

We define an event as representation of an incident and an event consists of two types 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 3.1 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 chapter, we aim to automatically detect this kind of causally related event pairs in tweets as illustrated in Fig. 3.1. 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 [7, 8] 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. [10], often fail to detect causal relationship between events in tweets. Hence, we propose an event context word extension technique to automatically detect causal

Table 3.2 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

relationship between candidate causal event e_1 and effect event e_2. Formally, the problem investigated in this chapter can be defined as follows:

$$f(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 \text{ causes } e_2, \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

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 chapter are given in Table 3.2.

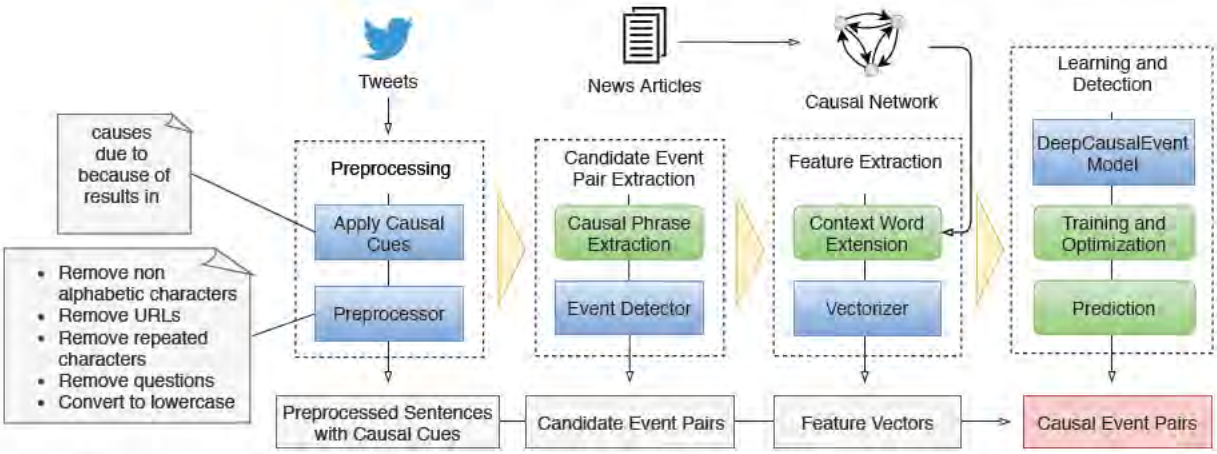


Fig. 3.2 An overview of our sequence-aware deep causal event detection model (adapted from Kayesh et al. [2])

3.3 Our Approach

In this chapter, 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 [2, 117], 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, 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. Fig. 3.2 displays a high-level overview of the workflow in the proposed approach.

Table 3.3 Causal cue words used for candidate causal and effect phrases extraction from tweets [2]

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

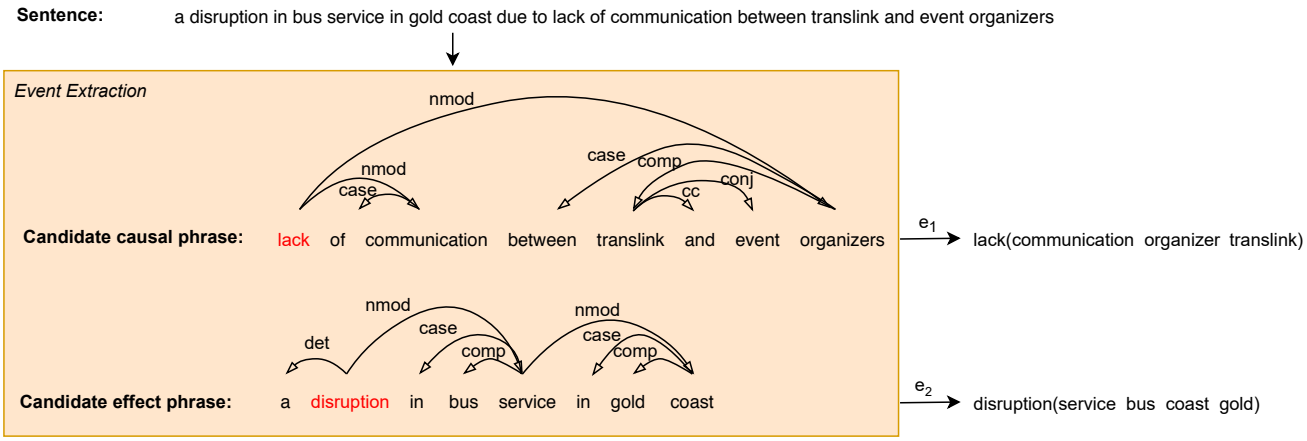


Fig. 3.3 An example of event pair extraction from a sentence [2]

3.3.1 Tweet Preprocessing

Tweet preprocessing is the first step in our 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 e.g., '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.

3.3.2 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 phrases. Table 3.3 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 $w_k^c(w_0^c, w_1^c, w_2^c...w_p^c)$ 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 $w_k^e(w_0^e, w_1^e, w_2^e...w_q^e)$ where w_k^e is the event keyword and w_q^e is an event context word. An event keyword 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 [118]. As proposed in Kayesh et al. [2], 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 work [2, 117], 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. Fig. 3.3 illustrates 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".

Table 3.4 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			

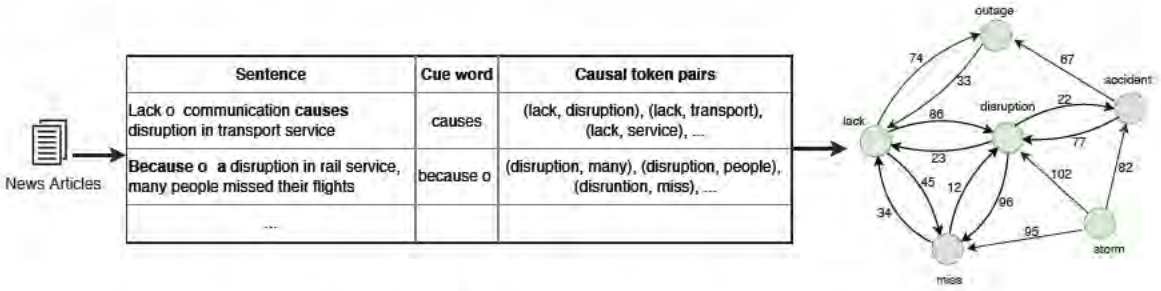


Fig. 3.4 Causal network construction from news articles [2]

3.3.3 Causal Network

Newspaper articles are a good source of causality related background knowledge [9,11,69, 71–73,111,119]. Hence, we capture background knowledge on causality and build a causal background knowledge network from a set of one million news articles[1] by following the technique proposed by Luo et al. [7]. The dataset is prepared by collecting news articles between the period of 1st November 2015 to 30 November 2015. More than 95K unique source websites are used to collect articles. On an average, an article in the dataset contains 405 words [120].

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 3.4. 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

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

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 3.3) to extract causal and effect phrases from newspaper 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, 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 illustrated in Fig. 3.4.

3.3.4 Context Word Extension

In this step, we discuss our technique to utilize background knowledge encoded in the causal network built in Section 3.3.3. We apply a context word extension technique [2] 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 $(w_0^c, w_1^c, w_2^c ... w_p^c)$ 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 $(w_0^e, w_1^e, w_2^e ... w_q^e)$ of e_2 by extracting the same number of effects of w_k^c from the network. A n-word extended candidate causal event $e_1 = \{w_k^c, (w_0^c, w_1^c, w_2^c ... w_p^c), (w_{ex_0}^c, w_{ex_1}^c, ... w_{ex_n}^c)\}$ and candidate effect event $e_2 = \{w_k^e, (w_0^e, w_1^e, w_2^e ... w_q^e), (w_{ex_0}^e, w_{ex_1}^e, ... w_{ex_n}^e)\}$ 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 3.1 and an example of context word extension where $n = 2$

Algorithm 3.1 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

is illustrated in Fig. 3.5.

3.3.5 Causal Event Detection

In this section we discuss the proposed DeepCausalEvent 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 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.

3.3.5.1 Vectorization

In this step, we vectorize our candidate events e_1 and e_2 into embedding vectors. We use a pretrained Word2vec model [121] to convert each words 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's 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 it's respective embedding vectors. We refer to the embedding vectors in e_1 as $\{v_k^c, (v_0^c, v_1^c, v_2^c ... v_p^c), (v_{ex_0}^c, v_{ex_1}^c, ... v_{ex_n}^c)\}$ and e_2 as $\{v_k^e, (v_0^e, v_1^e, v_2^e ... v_q^e), (v_{ex_0}^e, v_{ex_1}^e, ... v_{ex_n}^e)\}$. After the conversion of causal and effect events into vectors, they are sent to the deep causal event detection model.

3.3.5.2 The Proposed Deep Causal Event Detection Model

Our deep causal event detection model has two stages. In the first stage, we extract features from $e_1 = \{v_k^c, (v_0^c, v_1^c, v_2^c ... v_p^c), (v_{ex_0}^c, v_{ex_1}^c, ... v_{ex_n}^c)\}$ and $e_2 = \{v_k^e, (v_0^e, v_1^e, v_2^e ... v_q^e), (v_{ex_0}^e, v_{ex_1}^e, ... v_{ex_n}^e)\}$ separately and then combine the extracted features together. We extract the causal features by applying a BLSTM followed by a Multi-head Attention model on $(v_0^c, v_1^c, v_2^c ... v_p^c)$. Then we apply dense layers separately on v_k^c and $(v_{ex_0}^c, v_{ex_1}^c, ... v_{ex_n}^c)$. 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 Multi-head Attention model on $(v_0^e, v_1^e, v_2^e ... v_q^e)$ and then applying dense layers on v_k^e, and $(v_{ex_0}^e, v_{ex_1}^e, ... v_{ex_n}^e)$. 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 two-dimensional CNN models on the combined features to detect causality. Each CNN models are followed by a two dimensional 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.

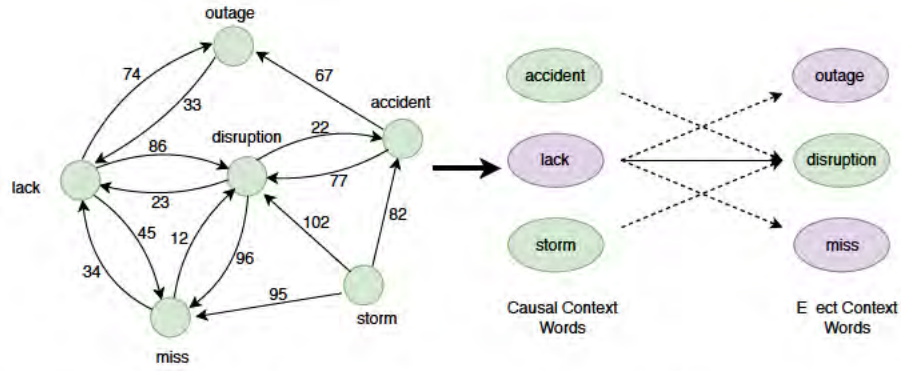


Fig. 3.5 An example of n-word context word extension, where $n = 2$ and the original candidate cause and effect keywords are *lack* and *disruption*, respectively [2]

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 Fig. 3.6.

3.4 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.

3.4.1 Dataset

We prepare our dataset by collecting more than 207k tweets using twitter API[2]. We only collect the tweets the are published within the date range between 5^{th} October 2017 and 7^{th} May 2018. We aim to collect the tweets that were related to the Commonwealth Games 2018 held in 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 3.3.1) and then we extract 913 pairs of candidate causal and effect

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

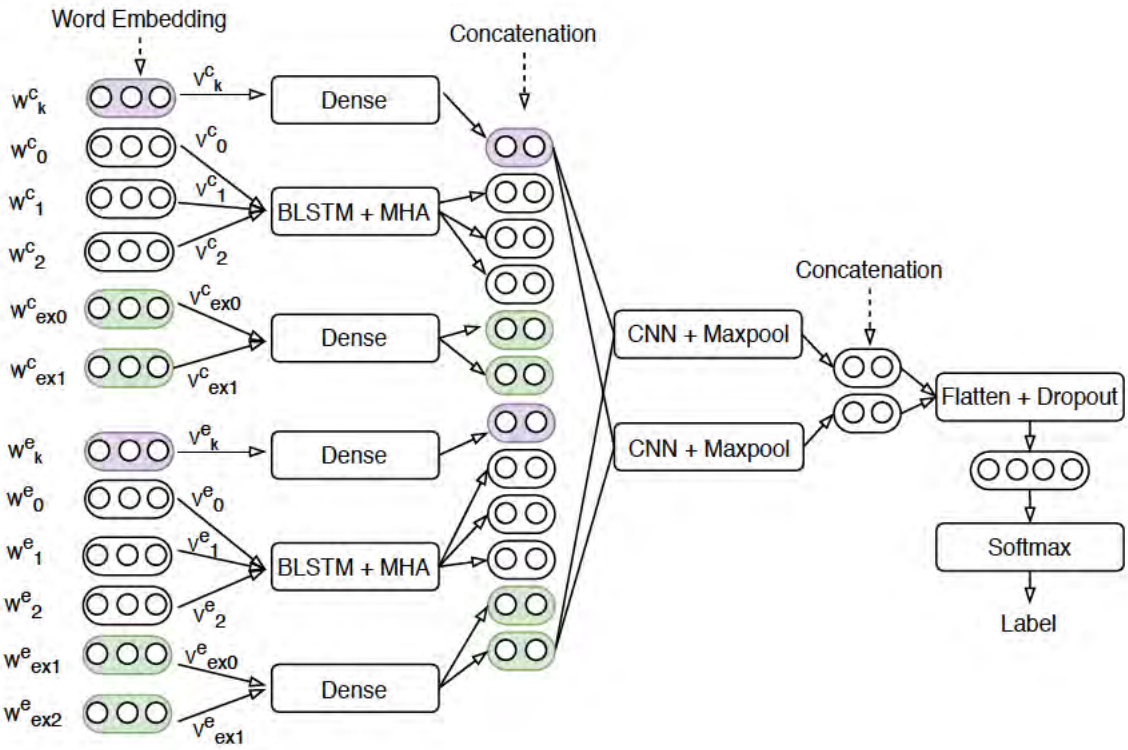


Fig. 3.6 Architecture of the proposed deep causal event detection model

events pairs by following the technique described in Section 3.3.2. 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 optimise 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 3.5.

3.4.2 Setup

We implement the proposed deep causal event detection model in python 3.6 language using the Keras [3] 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

[3]https://keras.io/

Table 3.5 Statistics of the tested dataset [2]

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

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 multi-head attention model. We use the same configuration for the BLSTM model and the multi-head 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. 3.2, where S is the maximum sequence length, Kr is the kernel size used in the CNN model. For the other 2D CNN model, 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 Eq. 3.2 to determine the pool size.

$$poolsize = (S - Kr + 1, 1) \tag{3.2}$$

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 4 epochs with batch size set to 32. We have finalised these parameters empirically. A summary of the model parameters are given in Table 3.6.

Table 3.6 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
Multi-head 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

3.4.3 Performance Evaluation

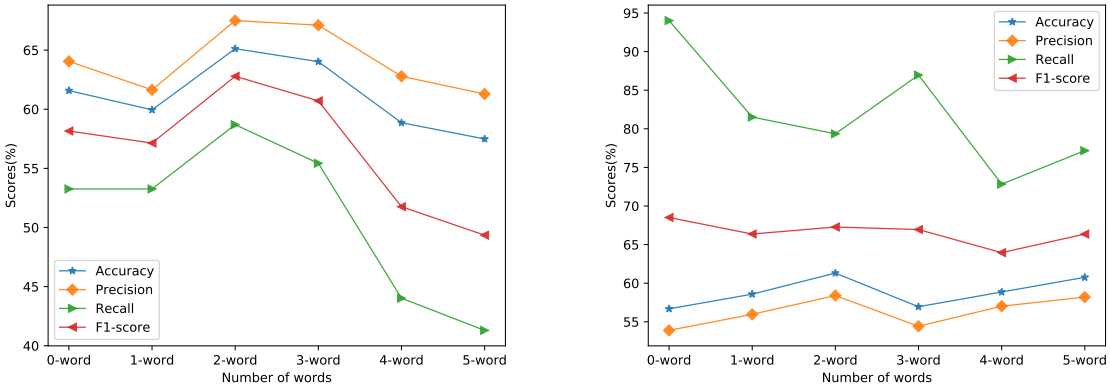
We compare our model's performance for different numbers of context word extensions while keeping our deep neural network model settings constant. The experiment's results are shown in Table 3.7 and Fig. 3.7. In our experiment, we discovered that the 2-word extension model has the highest accuracy (61.31%) and precision (58.04%) scores when compared to the other context word extensions.In addition, when compared to the 0-word or no word extension models, the 2-word extension model achieves 4.63 percent greater accuracy and 4.51 percent greater recall. The 0-word extension model's high recall (94.02%) and comparatively low precision (53.89%) suggest that the models tend to label most candidate causal events as causal regardless of the causal relationship between events. The rationale behind choosing the 2-word extension also evident from Fig. 3.8 which illustrates the ROC curve values for the same experiment settings. In Fig. 3.7, we can also see that a higher number of n in the n-word extension results in a rapid decline in the performance of *FFNN + n-word ext.* model compared to the DeepCausalEvent. The performance is more stable for DeepCausalEvent for the same number of word extensions. The reason behind this is the usage of more deep contextual features used in the proposed model compared to the *FFNN + n-word ext.* model. This makes the proposed model less dependent to the extended words.

Table 3.7 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

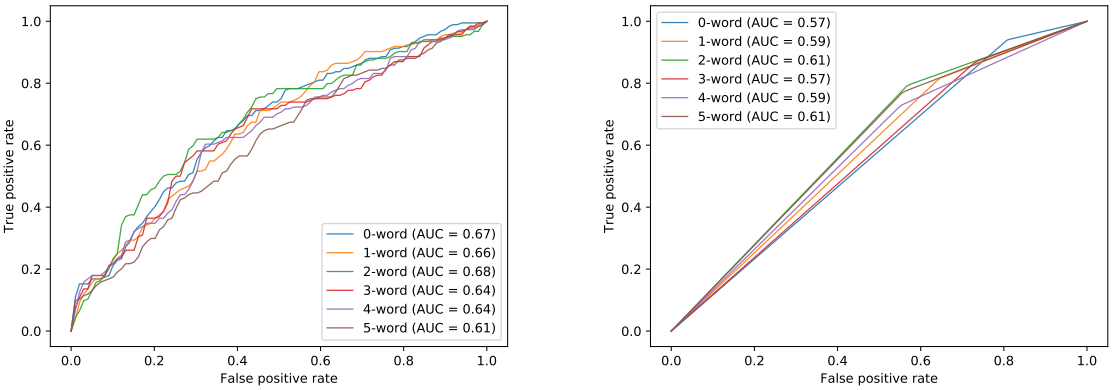
To demonstrate the effectiveness of the proposed model, we compare the proposed model DeepCausalEvent with a number of existing state-of-the-art approaches. We compare our approach with Luo et al.'s [7] commonsense-based approach (Commonsense), which applies commonsense knowledge to detect causality. This approach builds a causal network of commonsense and calculates causal strength between two phrases using that causal network. We also compare our approach with Sasaki et al. [8]'s *Commonsense + Multi-word*, that extends Luo et al.'s approach by proposing a multi-word casual network instead of single word network. Another benchmark approach (FFNN + Position) is a neural network appraoch proposed by Ponti et al. [10]. 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 [2] (*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 two-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 we compare our proposed model to the benchmark approaches in Table 3.8, we see that DeepCausalEvent outperforms all of the benchmark models in terms of recall and f1-score. Our proposed model has a recall of 79.35 percent and a f1-score of 67.28 percent. In our experiment, the best performer among the benchmark approaches is shown in Table



(a) FFNN + n-word ext. (b) DeepCausalEvent

Fig. 3.7 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. [2]) and (b) DeepCausalEvent model



(a) FFNN + n-word ext. (b) DeepCausalEvent

Fig. 3.8 The comparison of AUC values among different number of context word extensions: (a) FFNN+n-word ext. (adapted from Kayesh et al. [2]) and (b) DeepCausalEvent

3.8. Our model achieves lower accuracy and precision than the best benchmark model *FFNN + 2-word* . The low precision is the result of the low precision. Our proposed model improves recall by at least 17.39 percent and f1-score by at least 2.33 percent. This result demonstrates our approach's superiority over 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 3.8 illustrates the comparison of results among the proposed DeepCausalEvent model and other state-of-the-arts. The table shows that the commonsense-based techniques *Commonsense [7]* and *Commonsense + Multi-word [8]* achieves 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 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. Our proposed *DeepCausalEvent* model achieves the best recall and f1-scores 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.

3.4.4 Discussion

Table 3.9 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*

Table 3.8 Comparison of the proposed DeepCausalEvent method with existing approaches

Methods	Accuracy	Precision	Recall	F1-score
Commonsense [7]	50.95	56.67	9.24	15.89
Commonsense + Multi-word [8]	50.14	54.55	3.26	6.15
FFNN + Position [10]	59.40	60.12	56.52	58.26
FFNN + 2-word Ext. [2]	**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**

identifies it as non-causal. 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.

3.5 Summary

In this chapter, 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.

Table 3.9 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 + Multi-word**	**FFNN + Position**	**FFNN + 2-word-Ex**	**DeepCausalEvent**
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 #comonwealthgames2018-match	1	0	0	1	1	1
you cant 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 "an oficial"→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

STATEMENT OF CONTRIBUTION TO CO-AUTHORED PUBLISHED PAPER

Chapter 4 includes a co-authored journal paper, which has been published in Neurocomputing in 2022. The bibliographic details of the co-authored paper, including all authors, are:

- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, Ryoma Ohira, and Zhe Wang, "SCAN: A Shared Causal Attention Network for Adverse Drug Reactions Detection in Tweets.," Neurocomputing (2022), Volume 479, Pages 60-74.

My contribution to the paper involved: proposal of the SCAN model, implementation, experiments, writing and editing manuscript.

(Signed)  (Date) June 10, 2022

Humayun Kayesh

(Countersigned)  (Date) June 10, 2022

Corresponding author of paper: Md. Saiful Islam

(Countersigned)  (Date) June 10, 2022

Supervisor: Junhu Wang

Chapter 4

Causality for Adverse Drug Reactions Detection

Twitter is a popular social media site on which people post millions of tweets every day. As patients often share their experiences with drugs on Twitter, tweets can also be considered as a rich alternative source of *adverse drug reaction* (ADR)-related information. This information can be useful for health authorities and drug manufacturing companies to monitor the post-marketing effectiveness of drugs. However, the automatic detection of ADRs in tweets is challenging, as tweets are informal and prone to grammatical errors. The existing approaches to automatically detecting ADRs do not consider the cause-effect relationships between a drug and an ADR. In this chapter, we propose a novel shared causal attention network that exploits such cause-effect relationships to detect ADRs in tweets. In our approach, we split a tweet into the prefix, midfix, and postfix segments based on the position of the drug name in the tweet and separately extract causal features from the segments. We then share these separate causal features with both word and parts-of-speech features, and apply the multi-head self-attention mechanism. We run extensive experiments on three publicly available benchmark datasets to illustrate the effectiveness of the proposed approach.

4.1 Introduction

Adverse drug reactions (ADRs) are negative drug side effects [76]. Some people exhibit ADRs after taking prescribed medicine, and the ADRs may vary by age group. Hence, ADR detection and statistical data may help doctors minimize the clinical risks associated with ADRs and reduce healthcare costs for society when they are prescribing drugs [122]. The timely detection of ADRs is crucial and depends on an efficient ADR-reporting process.

The formal process of reporting ADRs includes lab tests performed by pharmaceutical companies before marketing the drug and the patient-reported post-marketing information of ADRs. However, both approaches have limitations. Lab tests are often performed in a simulated and controlled environment for a short period of time. Therefore, it is possible that some ADRs will not be identified by lab tests. To detect these ADRs, drug manufacturing companies open online portals to collect self-reported ADRs. However, patients are often not motivated enough to spend their valuable time on completing a long ADR reporting form. To tackle this problem, the automatic collection of ADRs from short text on social media platforms could be a viable alternative.

Tweets are posted on a wide variety of topics, including health, drugs, and medications. One study found that approximately 0.08% of tweets contain health-related information [123]. This data suggests the potential of social media platforms, such as Twitter, as a source of health-related information, as patients often share their health- and medication-related information with others on Twitter. Fig. 4.1 provides two examples of such tweets that mention unwanted reactions after taking drugs. In the first tweet, the user writes, "Ugh the Olanzapine makes me so tired :(," indicating that after taking *Olanzapine*, she feels tired. In the second tweet, the user writes, "Effexor has left me with the inability to cry," indicating that after taking *Effexor*, the user experienced difficulty with being unable to cry. By analyzing such tweets, we can identify cause-effect relationships between drugs and ADRs. In both tweets, the name of the drug can be considered to be the cause and the ADRs can be

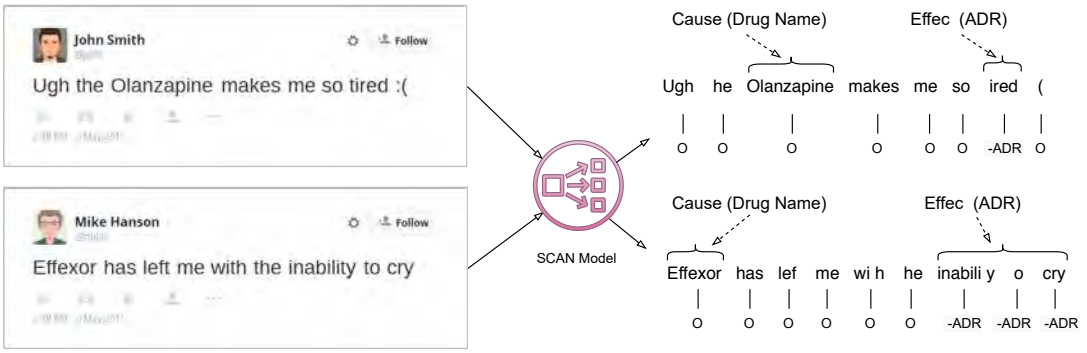


Fig. 4.1 ADRs in tweets with the cause-effect relationship

considered to be the associated effects. For example, in the first tweet, *Olanzapine* is the cause of being *tired*. In the second tweet, *Effexor* is the cause of the *inability to cry*. The automatic detection of such information from social media sites, such as Twitter, could provide a low-cost and reliable source of health information for both drug manufacturers and healthcare authorities.

Through a review of the literature, we only found a few existing approaches to detect ADRs in tweets [3, 4, 76, 124]. Yang et al. [76, 124] applied a word-occurrence based technique to detect ADRs correspond to the drug names in tweets. The main drawback of this approach is the lack of consistency. A drug name and an adverse reaction may frequently co-occur, but there is no guarantee that the drug is the cause of the adverse reactions. For example, "Thank god for vyvanse or I would be sleeping on the cash register rightzo now" contains a drug name (*vyvanse*) and a potential ADR word (*sleeping*). However, the word *sleeping* has not been used to describe as an adverse effect of the drug, and it should not be considered as an ADR in this tweet. Cocos et al. [4] proposed a technique that applies word embedding as the key feature to detect ADRs. However, this limits the generalizability of the system, as the model needs to be trained for the drug names. Hence, unlike word co-occurrence or embeddings-based approaches, we should emphasize the cause-effect relationships between the drug-ADR pairs in tweets.

Recently, we proposed a causality-driven sequence labeling approach to ADR detection in tweets [3]. This approach splits the tweet into the prefix, midfix, and postfix segments. The term frequency–inverse document frequency (tf–idf) features are then calculated for each segment and combined before applying a convolutional neural network (CNN)

followed by a multi-head self-attention (MHA) mechanism on the features. However, this approach results in overlapping features between the prefix, midfix, and postfix segments. In this chapter, we propose a novel shared causal attention network (SCAN) that detects ADRs by identifying causal relationships between drug names and ADR words in tweets [1]. The proposed SCAN model first extracts word features, parts-of-speech (POS) features, and causal features and then shares the causal features with the word and the POS features. It then applies a MHA layer to detect ADR words in tweets. Using a similar method to Kayesh et al. [3], the model splits every tweet into three segments: the prefix, midfix, and postfix, depending on the position of the drug name and the candidate ADR word, to extract causal features. However, in contrast with this research [3], we apply separate CNN layers to the prefix, midfix, and postfix segments to tackle the issue of overlapping features.

The key contributions of this chapter are as follows:

- we propose a novel SCAN that applies the MHA mechanism to the combination of causal features, word features, and POS features to detect ADRs in tweets;
- we propose an improved separate causal feature extraction strategy that solves the problem of overlapping features; and
- we perform extensive experiments on three publicly available benchmark datasets and compare our proposed method with other state-of-the-art approaches to show the effectiveness of the proposed model.

The remaining sections of this chapter are discussed as follows: the proposed ADR detection method is described in Section 4.2, the experimental results and discussions are presented in Section 4.3, and conclusions are drawn in Section 4.4.

[1] Our code is available at https://github.com/hkayesh/scan-neucom

4.2 Our Approach

In this section, we first formulate the research problem studied in this chapter and then describe our proposed SCAN model for ADR words detection in tweets.

4.2.1 Research Question

When a tweet contains an ADR and a drug name, we assume that there is a causal relationship between them, i.e, the *drug* causes the ADR mentioned in the tweet. We believe that this causal relation can be used to detect ADRs in tweets automatically. For example, "Ugh the Olanzapine makes me so tired :(" contains a drug name, *Olanzapine*, and an ADR word, *tired*, and there is a causal relationship between them. In this case, we can safely assume that the *Olanzapine* was the cause of the ADR. We intend to use such causal inferences to detect ADR words in tweets automatically. Our research question is:

RQ: *How can we automatically detect ADR words in tweets by applying a feature extraction technique that captures the cause-effect relationship between a drug and the corresponding ADRs?*

We assume that a set of drug names for which ADRs will be detected has already been chosen. The Twitter API is used to download a collection of tweets by sending the drug names as search keys. The locations of drug names are known in every tweet.

4.2.2 Problem Statement

We consider a tweet (τ) consisting of a sequence of words ($W = [w_1, w_2, ..., w_n]$), where n is the number of words in τ. Our aim is to develop a function (f) that takes τ as

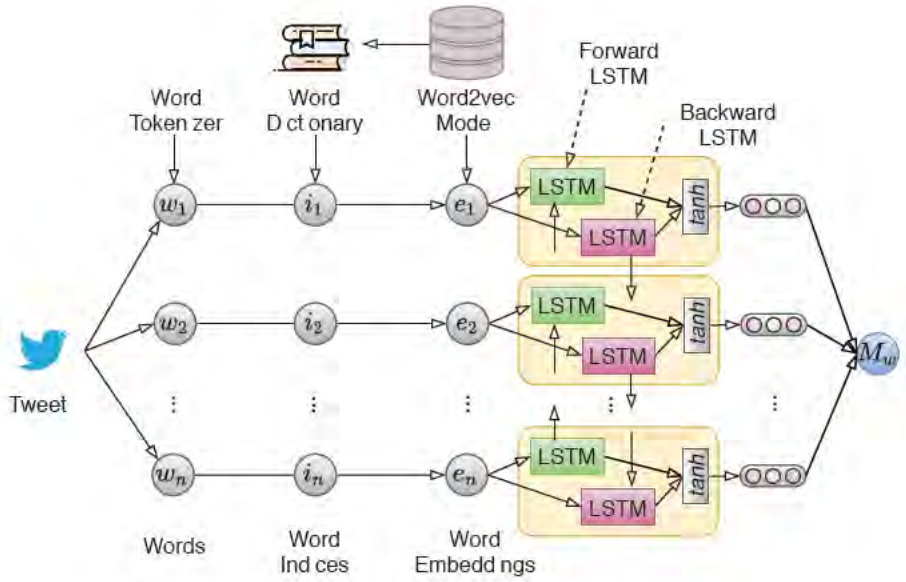


Fig. 4.2 Word features extraction from a tweet

the input and detects the ADR words. Therefore, f is a sequence labeling function that predicts the labels of each word in τ. The formal definition of f is given below.

$$f(W = \{w_i\}) \rightarrow L = \{l_i\}, i \in \{1, 2, .., n\} \tag{4.1}$$

$$\forall l_i \in L : l_i = \begin{cases} \text{I-ADR} & \text{if } w_i \text{ is an ADR word,} \\ \text{O} & \text{otherwise} \end{cases} \tag{4.2}$$

In the above equations, L is a sequence of the labels of the corresponding words in τ. The label 'I-ADR' (Inside-ADR) corresponds to an ADR word, and the label 'O' (Outside-ADR) corresponds to a non-ADR word. Fig. 4.1 shows the words in two example tweets and their corresponding labels that were detected by f.

4.2.3 Shared Causal Attention Network

Our model applies three types of features: word features, POS features, and causal features. The features are passed to our proposed SCAN model that uses a multi-head

self-attention mechanism to detect ADR words in tweets.

4.2.3.1 Word Features

We extract word embeddings to capture the useful semantic relationships between words. We use a publicly available pre-trained Word2vec model [125] that was trained on approximately 400 million domain-independent tweets.

- As a first step to convert τ into a sequence of word embeddings, we tokenize τ into words. We denote the sequence of words as $w_1, w_2, ..., w_n$, where n is the length of the sequence. We then use a dictionary of words to replace each word in the sequence with its corresponding index, which converts the word sequence into a sequence of indices, i.e., $i_1, i_2, ..., i_n$, to be used later to extract word embeddings. To avoid variable-length sequences, we use padding if a sequence is smaller than the longest sequence in the dataset. We use a pre-trained Word2vec [125] model to replace each word index with it's word embedding, and we denote the sequence of word embeddings as $e_1, e_2, ..., e_n$.

- We apply a BLSTM layer onto $e_1, e_2, ..., e_n$ to extract contextual word features. The layer applies a forward LSTM and a backward LSTM on the same input. We denote the last hidden state of the forward LSTM as $\overrightarrow{h_w}$ and the last hidden state of the backward LSTM as $\overleftarrow{h_w}$. We then take an average of the last hidden states as shown in Equation 4.3:

$$M_w = \overrightarrow{h_w} \| \overleftarrow{h_w} \tag{4.3}$$

  Where $M_w \in R^{n \times l}$ is the output of the BLSTM layer, n is the sequence length, and l is the feature length. The word feature extraction steps are illustrated in Fig. 4.2.

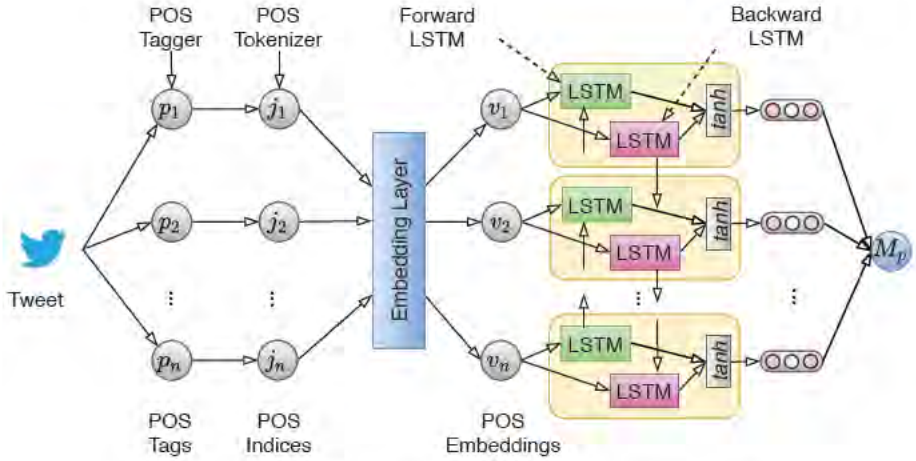


Fig. 4.3 POS feature extraction from a tweet

4.2.3.2 Parts-of-Speech Features

POS contains linguistic features that can be useful for ADR detection. A detailed description of our POS feature extraction is given below.

- First, we extract POS tags for each word in a tweet τ, and we denote the sequence of POS tags as $p_1, p_2, ..., p_n$, where n is the length of the sequence. We then use a POS tokenizer to convert $p_1, p_2, ..., p_n$ into a sequence of indices, $j_1, j_2, ..., j_n$, where each index corresponds to a POS tag. The tokenizer also applies padding to ensure the sequences are the same length. We pass $j_1, j_2, ..., j_n$ onto an embedding layer to convert the sequence of indices into a sequence of embeddings, and we denote the sequence of POS embeddings as $v_1, v_2, ..., v_n$. The values of the POS embeddings are learned during the training.

- To extract patterns from the POS embeddings, we apply the BLSTM layer. This layer applies two LSTMs (one forward and one backward). The last hidden state of the forward LSTM, $\overrightarrow{h_p}$, is averaged with the last hidden state of the backward LSTM, $\overleftarrow{h_p}$, as shown in Eq. 4.4. We use the activation function ($tanh$) in the BLSTM layer.

$$M_p = \overrightarrow{h_p} \| \overleftarrow{h_p} \tag{4.4}$$

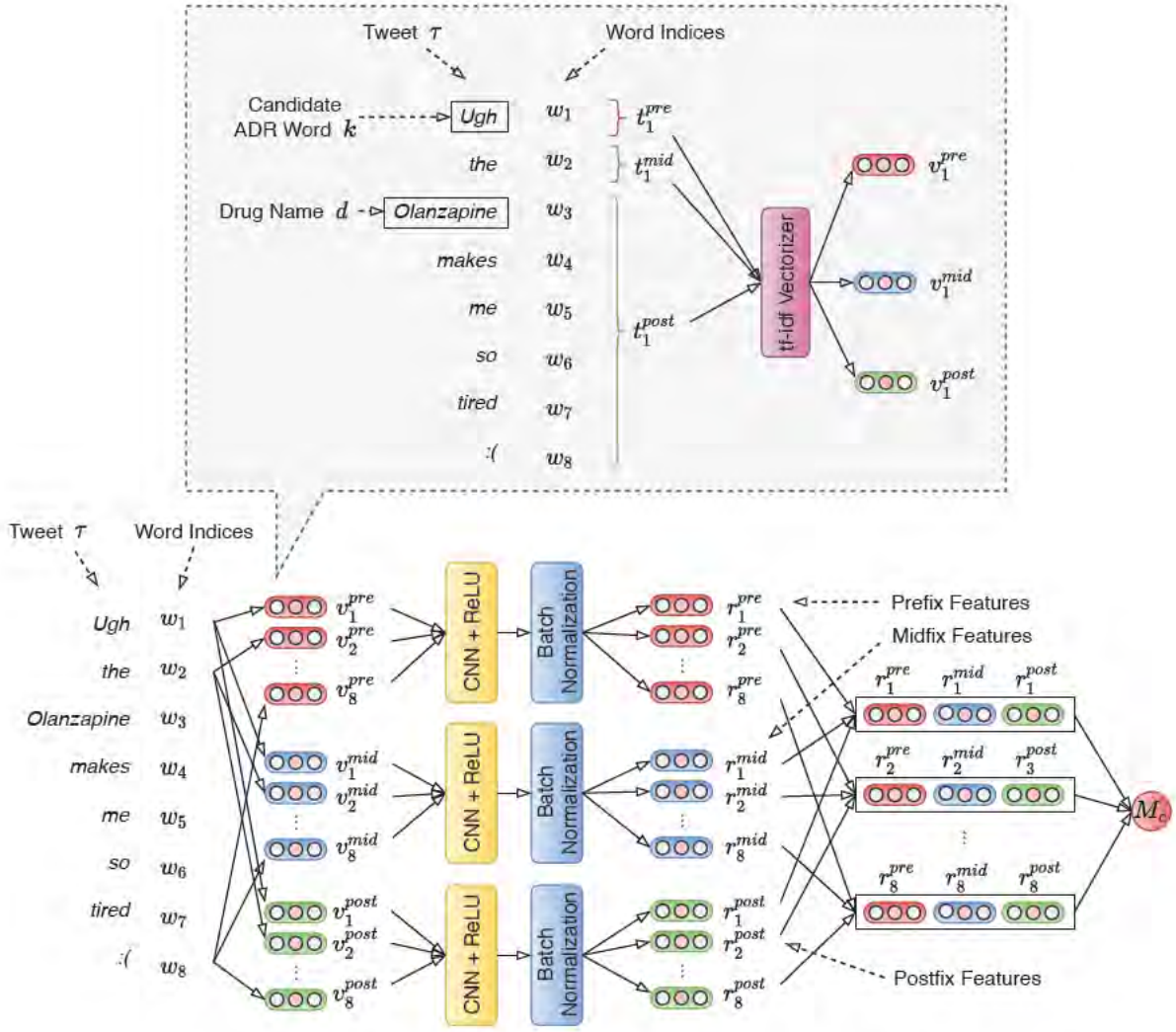


Fig. 4.4 Causal feature extraction from a tweet

Where $M_p \epsilon R^{n \times m}$ is the output from the BLSTM layer, and m is the size of the embeddings. An illustration of the POS feature extraction steps is shown in Fig. 4.3.

4.2.3.3 Causal Features

We assume that if a drug name (d) exists in τ then any of the other words in the tweet could be an ADR. Hence, we consider that d may have a cause-effect relationship with any of the other words in τ, where d is the candidate cause and the other words can be considered the candidate effects. In our proposed model, we extract causal features for each candidate ADR word in a tweet and share this feature with the word features and

POS features. To extract this causal feature, we split τ into three segments - a prefix, midfix, and postfix, considering each word (except the drug name) in τ as a candidate ADR word. It should be noted that each segment may have more than one word. The causal features used in the shared causal attention layer of the proposed SCAN model helps to focus attention on the correct contextual words while predicting the label of a candidate ADR word.

A detailed description of the proposed SCAN model is given below.

- For each candidate ADR word (k), we divide τ into the prefix, midfix, and postfix. From the beginning of τ, the words until k or d (whichever appears first) are considered to be the prefix, words between k and d are considered to be the midfix, and the words starting from k or d (whichever appears last) until the end of τ are considered to be the postfix. Let's assume that τ has a sequence of words, $w_1, w_2, ..., w_n$, and w_3 is a drug name. To extract vectors for a candidate ADR word (w_1), we split $w_1, w_2, ..., w_n$ into the prefix (t_1^{pre}), midfix (t_1^{mid}) and postfix (t_1^{post}), where $t_1^{pre} = [w_1]$, $t_1^{mid} = [w_2]$, and $t_1^{post} = [w_3, w_4, ..., w_n]$. Then we apply a tf-idf vectorizer to convert t_1^{pre}, t_1^{mid}, and t_1^{post} into vectors. We denote the corresponding vectors as v_1^{pre}, v_1^{mid}, and v_1^{post}, respectively. Similarly, for each word w_i, we extract three vectors, v_i^{pre}, v_i^{mid}, and v_i^{post}.

- After converting the prefix, midfix, and postfix segments for each word in a tweet τ into vectors, we extract causal features from the vectors. The prefix features are extracted from the collection of all the prefix vectors, $v_1^{pre}, v_2^{pre}, ..., v_n^{pre}$. We apply a one-dimensional CNN to $v_1^{pre}, v_2^{pre}, ..., v_n^{pre}$ followed by a fully connected layer with rectified linear unit (ReLU) as an activation function. To avoid over-fitting to the training data and achieve better generalization, we apply a batch normalization layer to it and the output of this layer is denoted as $r_1^{pre}, r_2^{pre}, ..., r_n^{pre}$. Similarly, we extract the midfix features, $r_1^{mid}, r_2^{mid}, ..., r_n^{mid}$, and the postfix features, r_1^{post}, $r_2^{post}, ..., r_n^{post}$, for each word.

- Unlike Kayesh et al. [3], we extract separate causal features from the tf-idf vectors of the prefix, midfix, and postfix segments for each word instead of combining them

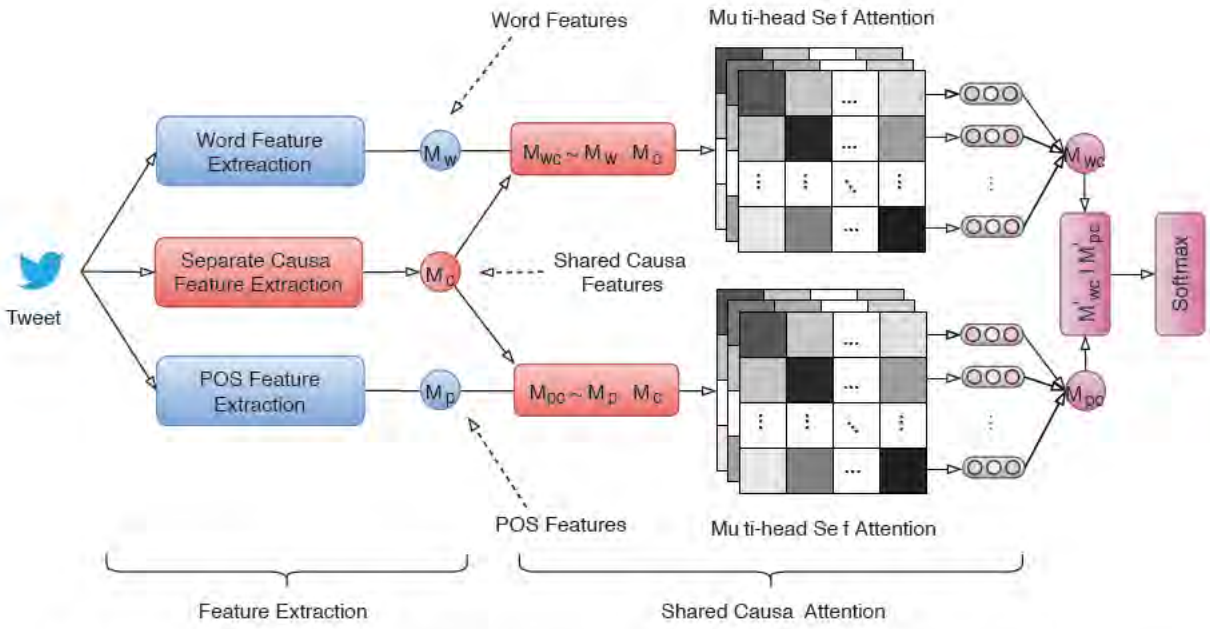


Fig. 4.5 The architecture of the proposed SCAN model for ADR words detection

before feature extraction. This technique reduces the overlapping of features among the prefix, midfix, and postfix segments. Another improvement that we make in our causal feature extraction is that we place a greater emphasis on the midfix features rather than treating each segment equally. We assume that the midfix segment contains more distinguishable features when compared with the prefix and postfix segments. The reason for this that it was observed that prefix or midfix segments were often the same for two subsequent words, depending on the position of d in τ, but the midfix segment was always different.

- The features extracted from the segments for each word are then combined. For example, the combined features for w_1 in τ can be represented as the combination of r_1^{pre}, r_1^{mid}, and r_1^{post}. Similarly, we combine features for each word, and the combined feature matrix for τ is denoted as M_c. An illustration of this causal feature extraction technique is displayed in Fig. 4.4.

4.2.3.4 Network Architecture

We share the causal features, M_c, with M_w and M_p in the attention layer of the model. We combine M_c and M_w and denote the output as M_{cw}, as shown in Eq. 4.5. Similarly, M_{cp} represents the combination of M_c and M_p, as shown in Eq. 4.6. Then we apply two separate MHA layers to M_{wc} and M_{pc}. The outcomes of the attention layers, M'_{wc} and M'_{pc} are then combined before being passed to the Softmax layer for label prediction.

$$M_{wc} = (M_w \,|\, M_c) \tag{4.5}$$

$$M_{pc} = (M_p \,|\, M_c) \tag{4.6}$$

The complete architecture and workflow of the proposed SCAN model for ADR words detection in tweets is illustrated in Fig 4.5.

4.3 Experimental Evaluation and Analysis

In this section, we discuss the experimental results and demonstrate the effectiveness of the proposed approach in ADR words detection in tweets.

4.3.1 Dataset Setup

In this experiment, we used three publicly available human-annotated benchmark datasets. We also combined these datasets to prepare a fourth dataset. A brief description of the datasets is given below.

- **ASU_CHOP Dataset:** This dataset was shared by Cocos et al. [4, 126]. The authors updated the Twitter ADR Dataset (v1.0), which was originally published

by Nikfarjam et al. [93,127]. The dataset contains tweet IDs and annotations but no text. Therefore, we downloaded the dataset using the script provided by Cocos et al. [4]. At the time of download, we only found 664 unique tweets available online. Hence, the dataset used in our experiment contains a total of 791 examples and 664 unique tweets. We also excluded the non-ADR annotations, such as indications from the dataset to focus only on the ADR word detection. As a result, we could not directly compare our results with the results published by Cocos et al. [4]. Instead, we run their proposed model proposed against our downloaded dataset and compared its performance against our model.

- **Social Media Mining for Health Applications (SMM4H) Dataset:** This dataset is published by SMM4H's workshop 2019[2], shared task 2. This dataset contains a total of 1983 examples and 1832 unique tweets.

- **WEB_RADR Dataset:** This dataset was published by Dietrich et al. [128] and originally contained more than 57k tweets but only 1057 tweets are labeled as having adverse events. Before using it in our experiment, we extracted the tweets with adverse events and performed preprocessing, such as removing the tweets that contained more than one drug name, to make it consistent with the other two benchmark datasets. After preprocessing, the dataset used in our experiment contained 748 tweets.

- **Combined Dataset:** This dataset is the combination of the three datasets discussed above and contains 2421 unique tweets.

As we were only focusing on ADR word detection, we removed any word labels other than 'O' and 'I-ADR' from the datasets and replaced them with the non-ADR label, 'O'. For each dataset, we used 75% of the data to train the models and the remaining 25% data for the test. Of the training data, 90% was used for training and 10% for validation and parameter optimization. Table 4.1 presents a summary of the dataset statistics for our benchmark datasets.

[2]https://healthlanguageprocessing.org/smm4h19/

Table 4.1 Dataset Statistics

Dataset	Tweets(ADRs) Training Set	Test Set
ASU_CHOP Dataset	585(492)	206(172)
SMM4H Dataset	1487(1368)	496(464)
WEB_RADR Dataset	561(561)	187(187)
Combined	2633(2421)	889(823)

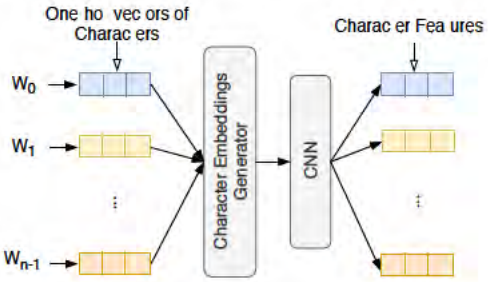
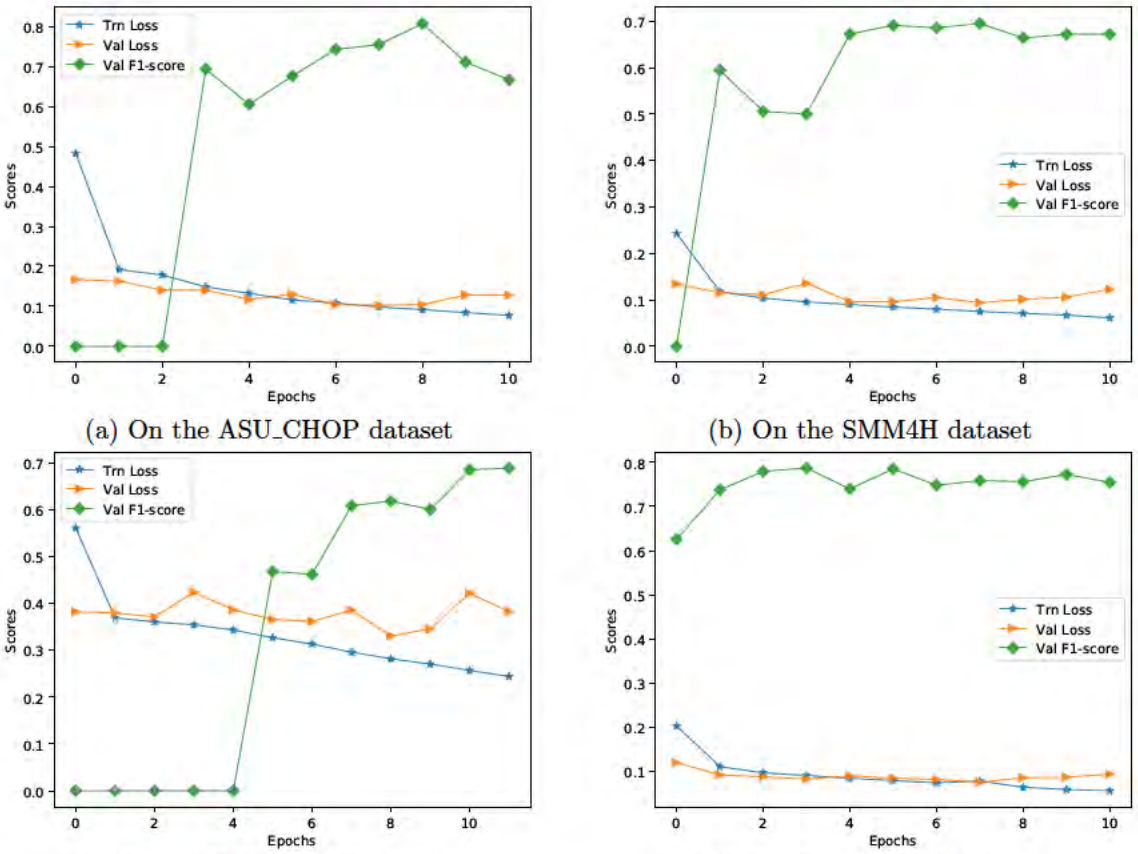


Fig. 4.6 Generating character features from each word in a tweet [3]

4.3.2 Benchmark Methods

In our experiments, we compared our model against several benchmark and state-of-the-art models, such as CRF [25], Cocos et al. [4] and CausalMHA [3]. A brief description of all the models that were experimented on in this work is given below.

- **CRF** [25]: The CRF model is applied to the word-level features. The features consist of pretrained word embeddings.
- **Cocos et al.** [4]: Unlike the previous approach, this approach only applies a BLSTM to the word embeddings. The same pre-trained word embeddings were used on this model as on the previous model.
- **CausalBLSTM**: In this model, we applied a BLSTM layer to the word embedding features. The word features were combined with the character embeddings and causal features. The character embeddings consisted of 100 dimensions, and we extracted the character features by applying a CNN layer to the character embeddings. An illustration of the character feature generation is shown in Fig. 4.6. After extracting the word features, character features, and



(a) On the ASU_CHOP dataset

(b) On the SMM4H dataset

(c) On the WEB_RADR dataset

(d) On the combined dataset

Fig. 4.7 Optimization of loss function in our SCAN model for ADR detection in tweets

causal features, we combined them and passed them to a dense layer to detect the word labels.

- **CharMHA**: This benchmark model combines the word and character features, then passes them to an MHA layer. We followed the same approach for character feature extraction as illustrated in Fig. 4.6.

- **CharCausalMHA**: This approach combines the word, character and causal features before applying an MHA layer to detect ADR labels.

- **MHA**: This model applies an MHA layer to the word features. The word features are extracted by applying a BLSTM layer to the word embeddings. No causal features are used in this model.

- **CausalMHA**: This model combines the word and causal features before applying an MHA layer and is our previous model [3] for ADR detection in tweets.

- **Word + POS**: In this model, we removed the causal features from the inputs and excluded the SCAN to measure the performance of the model without the causal features and the attention network.
- **Word Only**: This model uses only the word features to detect ADRs in tweets. We removed the causal and POS features from the SCAN model to illustrate the effectiveness of the features.

4.3.3 Experimental Settings

In the word feature extraction step, to convert each word into vectors, we used a pre-trained Word2vec [125] model that was trained on a domain-independent Twitter dataset. The word embeddings consisted of 400 dimensions, and each of the two LSTM models applied to the word embeddings had 80 units. In the POS feature extraction step, we used an embedding size of 16, and each LSTM model used to extract the POS features had 20 units. The activation function used in the LSTMs was *tanh*, and the dropouts were set to 0.2. In the causal features extraction steps, we used one-dimensional CNNs with 80 filters and a kernel size of 5. To avoid over-fitting we used batch normalization layers with the ReLU activation function. The maximum sequence length used to train the model was 41. We also used the *RMSprop* optimizer [129] to optimize *accuracy*, using the *categorical crossentropy* loss function. For benchmarking and evaluation, we calculated the approximate match [130, 131] score, and report precision, recall, and F1-scores. In the approximate match score calculation, if the predicted ADR words were a substring of the human-annotated ADR words (or vice versa), it was considered to be correct. The approximate match score was used to evaluate sequence labeling tasks in ADR detection [3, 4, 93].

In our experiments, we trained and evaluated each model for 30 runs using different random seeds to minimize the effect of randomness in the results. We reported the average of the top one, five, and ten F1-scores. In each run, we trained our model using the training data and optimized the loss function using the validation data. We

Table 4.2 Experimental results on the ASU_CHOP dataset

	Model	Precision	Recall	F1-score
Top 1	CRF [25]	**0.8824**	0.4688	0.6122
	Cocos et al. [4]	0.7189	0.8313	**0.7710**
	CausalBLSTM	0.7770	0.7188	0.7468
	CharMHA	0.6748	0.8688	0.7596
	CharCausalMHA	0.8235	0.7000	0.7568
	MHA	0.7440	0.7813	**0.7622**
	CausalMHA	0.6636	**0.8875**	0.7594
	SCAN	0.7470	0.7750	**0.7607**
Top 5	CRF [25]	**0.8732 ± 0.0167**	0.4375 ± 0.0164	0.5827 ± 0.0149
	Cocos et al. [4]	0.7213 ± 0.0338	0.7988 ± 0.0461	**0.7561 ± 0.0112**
	CausalBLSTM	0.7209 ± 0.0737	0.7338 ± 0.0734	0.7200 ± 0.0182
	CharMHA	0.7159 ± 0.0573	0.7813 ± 0.0747	0.7413 ± 0.0110
	CharCausalMHA	0.6915 ± 0.0672	0.8013 ± 0.0576	0.7371 ± 0.0124
	MHA	0.7346 ± 0.0224	0.7725 ± 0.0284	**0.7522 ± 0.0075**
	CausalMHA	0.6832 ± 0.0271	**0.8425 ± 0.0394**	**0.7531 ± 0.0044**
	SCAN	0.7155 ± 0.0200	0.7825 ± 0.0131	**0.7471 ± 0.0076**
Top 10	CRF [25]	**0.8784 ± 0.0089**	0.4275 ± 0.0102	0.5749 ± 0.0088
	Cocos et al. [4]	0.7143 ± 0.0257	0.7825 ± 0.0329	**0.7445 ± 0.0094**
	CausalBLSTM	0.7045 ± 0.0495	0.7119 ± 0.0464	0.7014 ± 0.0153
	CharMHA	0.6974 ± 0.0445	0.7788 ± 0.0551	0.7284 ± 0.0101
	CharCausalMHA	0.7024 ± 0.0598	0.7694 ± 0.0695	0.7214 ± 0.0120
	MHA	0.7601 ± 0.0373	0.7331 ± 0.0442	**0.7412 ± 0.0085**
	CausalMHA	0.7043 ± 0.0307	**0.7994 ± 0.0438**	**0.7447 ± 0.0062**
	SCAN	0.6909 ± 0.0304	**0.7988 ± 0.0300**	**0.7381 ± 0.0072**

kept training and saving the intermediate model checkpoints until the model started to overfit the training data. By investigating the performance of the model on the validation data, we chose the best model and applied it to the test data. Fig. 4.7 illustrates the loss function optimization scores on the ASU_CHOP, SMM4H, WEB_RADR, and the combined dataset.

Our proposed model contains $5,688,073$ parameters. The training and prediction time of our model is 1.40 seconds and the 0.02 second per tweet, respectively, on the combined dataset. We calculated the training time by taking the sum of the duration of each epoch. We ran the experiment 30 times and took an average to calculate the training and prediction times. We ran our experiments on a Linux machine that runs Ubuntu 18.04 LTS. The computer has an Intel Core i7-7700 (3.60GHz) 8-core processor and 32GB of RAM.

Table 4.3 Experimental results on the SMM4H dataset

	Model	Precision	Recall	F1-score
Top 1	CRF [25]	**0.5452**	0.4342	0.4834
	Cocos et al. [4]	0.4748	0.8660	0.6134
	CausalBLSTM	0.4689	0.9156	0.6202
	CharMHA	0.5261	0.8238	**0.6422**
	CharCausalMHA	0.4759	**0.9330**	**0.6303**
	MHA	0.4957	0.8610	**0.6292**
	CausalMHA	0.5053	0.8313	**0.6285**
	SCAN	0.4950	0.8561	**0.6273**
Top 5	CRF [25]	**0.5535 ± 0.0076**	0.4159 ± 0.0121	0.4747 ± 0.0052
	Cocos et al. [4]	0.5126 ± 0.0219	0.7305 ± 0.0677	0.5998 ± 0.0072
	CausalBLSTM	0.5202 ± 0.0267	0.7454 ± 0.0898	0.6085 ± 0.0097
	CharMHA	0.5211 ± 0.0046	0.7846 ± 0.0200	**0.6262 ± 0.0088**
	CharCausalMHA	0.4859 ± 0.0303	**0.8531 ± 0.0957**	0.6148 ± 0.0111
	MHA	0.4984 ± 0.0231	0.8347 ± 0.0632	**0.6220 ± 0.0051**
	CausalMHA	0.5315 ± 0.0237	0.7509 ± 0.0556	**0.6202 ± 0.0049**
	SCAN	0.5307 ± 0.0195	0.7529 ± 0.0545	**0.6207 ± 0.0040**
Top 10	CRF [25]	**0.5573 ± 0.0047**	0.4094 ± 0.0071	0.4719 ± 0.0031
	Cocos et al. [4]	0.5314 ± 0.0179	0.6700 ± 0.0521	0.5888 ± 0.0082
	CausalBLSTM	0.5217 ± 0.0145	0.7129 ± 0.0500	0.5995 ± 0.0078
	CharMHA	0.5242 ± 0.0206	0.7355 ± 0.0488	0.6085 ± 0.0133
	CharCausalMHA	0.5019 ± 0.0257	0.7777 ± 0.0786	0.6025 ± 0.0097
	MHA	0.4975 ± 0.0156	**0.8146 ± 0.0462**	**0.6152 ± 0.0055**
	CausalMHA	0.5425 ± 0.0185	0.7117 ± 0.0415	**0.6128 ± 0.0056**
	SCAN	0.5327 ± 0.0144	0.7323 ± 0.0341	**0.6150 ± 0.0045**

4.3.4 Results and Analysis

4.3.4.1 Comparison with Competing Models

The experimental results on the four benchmark datasets are highlighted in Table 4.2 - 4.5 and a summary of the improvement rates achieved by the MHA, CausalMHA, and SCAN models against the model proposed by Cocos et al. [4] is shown in Table 4.6. In our experiments, the CRF model achieved the lowest f1-scores across all the datasets. However, the MHA, CausalMHA and SCAN performs consistently on all four datasets.

The summarized results displayed in Table 4.6 show that our proposed SCAN model outperforms the model proposed by Cocos et al. in three out of the four datasets. When considering only the top F1-scores, the greatest improvement rate was 6.53% on the WEB_RADR dataset. It was found that the model proposed by Cocos et al. performs well on smaller datasets, such as ASU_CHOP that only has 492 and 172 unique tweets

Table 4.4 Experimental results on the WEB_RADR dataset

	Model	Precision	Recall	F1-score
Top 1	CRF [25]	0.7833	0.2597	0.3900
	Cocos et al. [4]	**0.5511**	0.6851	0.6108
	CausalBLSTM	0.5378	0.6685	0.5961
	CharMHA	0.4696	0.8122	0.5951
	CharCausalMHA	0.4735	0.7403	0.5776
	MHA	0.5468	0.8066	**0.6518**
	CausalMHA	0.4940	**0.9116**	0.6408
	SCAN	0.5094	0.9006	**0.6507**
Top 5	CRF [25]	**0.7872 ± 0.0277**	0.2398 ± 0.0111	0.3674 ± 0.0135
	Cocos et al. [4]	0.5342 ± 0.0125	0.6840 ± 0.0291	0.5994 ± 0.0097
	CausalBLSTM	0.5161 ± 0.0374	0.6685 ± 0.0789	0.5773 ± 0.0128
	CharMHA	0.4973 ± 0.0320	0.6906 ± 0.0823	0.5735 ± 0.0120
	CharCausalMHA	0.4850 ± 0.0169	0.6994 ± 0.0241	0.5722 ± 0.0087
	MHA	0.5130 ± 0.0255	0.8619 ± 0.0686	**0.6407 ± 0.0101**
	CausalMHA	0.5095 ± 0.0112	0.8144 ± 0.0546	0.6257 ± 0.0106
	SCAN	0.5144 ± 0.0253	**0.8420 ± 0.0843**	**0.6351 ± 0.0112**
Top 10	CRF [25]	**0.7921 ± 0.0212**	0.2287 ± 0.0096	0.3545 ± 0.0110
	Cocos et al. [4]	0.5316 ± 0.0165	0.6602 ± 0.0332	0.5870 ± 0.0095
	CausalBLSTM	0.5130 ± 0.0198	0.5862 ± 0.0666	0.5420 ± 0.0259
	CharMHA	0.4978 ± 0.0207	0.6492 ± 0.0576	0.5584 ± 0.0130
	CharCausalMHA	0.4809 ± 0.0150	0.6343 ± 0.0457	0.5451 ± 0.0197
	MHA	0.5241 ± 0.0275	**0.8127 ± 0.0806**	**0.6294 ± 0.0089**
	CausalMHA	0.5131 ± 0.0120	**0.7702 ± 0.0428**	**0.6140 ± 0.0093**
	SCAN	0.5272 ± 0.0171	0.7425 ± 0.0793	**0.6102 ± 0.0178**

in the training and test sets, respectively. However, the MHA, CausalMHA, and SCAN performed comparatively better on larger datasets, such as the combined dataset that has 2421 and 823 unique tweets in the training and test sets, respectively.

A model with higher recall and acceptable precision is preferable to one with high precision but low recall in the ADR detection task. A model with high recall and moderate precision can detect a wide range of ADRs, but some incorrect ADRs may be detected by additional lab tests. A high precision, low recall ADR detection model, on the other hand, may detect only a subset of the actual ADRs. Hence, we optimised the proposed SCAN model to achieve a high recall while keeping the precision at an acceptable rate. The experimental results on the combined dataset are shown in Table 4.5. It demonstrates that the model proposed by Cocos et al. achieves a higher recall when only the top F1-score is considered, but the SCAN model achieves a higher recall when the top ten F1-scores are averaged.

Overall, we find that for ADR detection in tweets the MHA, CausalMHA, and SCAN

Table 4.5 Experimental results on the combined dataset

	Model	Precision	Recall	F1-score
Top 1	CRF [25]	**0.6196**	0.4600	0.5280
	Cocos et al. [4]	0.5725	0.7993	0.6672
	CausalBLSTM	0.5812	0.7371	0.6500
	CharMHA	0.5877	0.7798	0.6702
	CharCausalMHA	0.5915	0.7869	**0.6753**
	MHA	0.5814	0.7993	**0.6731**
	CausalMHA	0.5772	**0.8099**	**0.6741**
	SCAN	0.5995	0.7922	**0.6825**
Top 5	CRF [25]	0.6185 ± 0.0037	0.4515 ± 0.0078	0.5219 ± 0.0059
	Cocos et al. [4]	0.5857 ± 0.0086	0.7510 ± 0.0258	0.6577 ± 0.0050
	CausalBLSTM	0.5795 ± 0.0219	0.7339 ± 0.0337	0.6465 ± 0.0029
	CharMHA	0.5863 ± 0.0235	0.7719 ± 0.0543	0.6644 ± 0.0049
	CharCausalMHA	**0.5891 ± 0.0123**	0.7776 ± 0.0258	**0.6699 ± 0.0032**
	MHA	0.5771 ± 0.0172	0.7886 ± 0.0364	**0.6656 ± 0.0056**
	CausalMHA	0.5736 ± 0.0209	**0.8004 ± 0.0400**	**0.6670 ± 0.0045**
	SCAN	0.5879 ± 0.0339	0.7780 ± 0.0587	**0.6667 ± 0.0079**
Top 10	CRF [25]	**0.6169 ± 0.0021**	0.4437 ± 0.0064	0.5161 ± 0.0048
	Cocos et al. [4]	0.5877 ± 0.0156	0.7396 ± 0.0292	0.6534 ± 0.0039
	CausalBLSTM	0.5882 ± 0.0142	0.7112 ± 0.0251	0.6426 ± 0.0031
	CharMHA	0.5803 ± 0.0161	0.7684 ± 0.0327	**0.6594 ± 0.0042**
	CharCausalMHA	0.5936 ± 0.0068	0.7549 ± 0.0201	**0.6640 ± 0.0043**
	MHA	0.5711 ± 0.0126	0.7755 ± 0.0245	**0.6568 ± 0.0065**
	CausalMHA	0.5812 ± 0.0222	0.7700 ± 0.0434	**0.6591 ± 0.0061**
	SCAN	0.5770 ± 0.0254	**0.7840 ± 0.0446**	**0.6609 ± 0.0055**

Table 4.6 Average improvement in F1-score (%) achieved by the CausalMHA and SCAN models against the model proposed by Cocos et al. [4]

	Top 1			Top 5			Top 10		
Dataset	**MHA**	**CausalMHA**	**SCAN**	**MHA**	**CausalMHA**	**SCAN**	**MHA**	**CausalMHA**	**SCAN**
ASU_CHUP	-1.14	-1.50	-1.34	-0.51	-0.40	-1.19	-0.44	0.03	-0.86
SMM4H	2.57	2.46	2.27	3.69	3.40	3.48	4.48	4.08	4.45
WEB_RADR	6.71	4.91	6.53	6.90	4.39	5.96	7.22	4.60	3.95
Combined	0.89	1.03	2.29	1.19	1.41	1.37	0.53	0.87	1.15

achieved comparable performance on small to moderately large datasets. Each of the aforementioned models contains MHA, and therefore, it can be argued that MHA is an important technique to use in a sequence labeling task such as as ADR word detection.

4.3.4.2 The Effectiveness of the Shared Causal Attention Layer

We performed an ablation study to evaluate the effectiveness of the proposed SCAN model. We implemented a model that only uses word features and another model that only uses the word and POS features. We aim to show how these models perform without our proposed shared causal attention layer, and Table 4.7 - 4.10 highlight the results of the

Table 4.7 Results of the ablation study on the ASU_CHOP dataset

	Model	Precision	Recall	F1-score
Top 1	Word Only	0.7158	0.8188	0.7638
	Word + POS	0.7427	0.7938	0.7674
	SCAN	0.7470	0.7750	0.7607
Top 5	Word Only	0.7339 ± 0.0235	0.7763 ± 0.0286	0.7536 ± 0.0056
	Word + POS	0.7590 ± 0.0637	0.7663 ± 0.0545	0.7568 ± 0.0057
	SCAN	0.7155 ± 0.0200	0.7825 ± 0.0131	0.7471 ± 0.0076
Top 10	Word Only	0.7286 ± 0.0297	0.7731 ± 0.0315	0.7474 ± 0.0063
	Word + POS	0.7663 ± 0.0393	0.7356 ± 0.0409	0.7460 ± 0.0078
	SCAN	0.6909 ± 0.0304	0.7988 ± 0.0300	0.7381 ± 0.0072

Table 4.8 Results of the ablation study on the SMM4H dataset

	Model	Precision	Recall	F1-score
Top 1	Word Only	0.4906	0.8387	0.6190
	Word + POS	0.5291	0.7444	0.6186
	SCAN	0.4950	0.8561	0.6273
Top 5	Word Only	0.5230 ± 0.0173	0.7419 ± 0.0517	0.6119 ± 0.0045
	Word + POS	0.5248 ± 0.0050	0.7315 ± 0.0137	0.6111 ± 0.0058
	SCAN	0.5307 ± 0.0195	0.7529 ± 0.0545	0.6207 ± 0.0040
Top 10	Word Only	0.5290 ± 0.0114	0.7156 ± 0.0322	0.6069 ± 0.0040
	Word + POS	0.5254 ± 0.0066	0.7151 ± 0.0174	0.6054 ± 0.0047
	SCAN	0.5327 ± 0.0144	0.7323 ± 0.0341	0.6150 ± 0.0045

ablation study on the four individual datasets. The improvement rates of SCAN against the word only and word + POS models are summarized in Table 4.11. The results suggest that the model with the shared causal attention layer performed better when compared with the word only and word + POS models. Table 4.11 suggests that applying SCAN can achieve better F1-scores on the SMM4H, WEB_RADR, and combined datasets. The improvement rate ranged from 1.33% to 3.18% if we consider only the top F1-scores, 0.52% to 2.96% for the top five F1-scores, and 0.2% to 2.1% for the top ten F1-scores. This demonstrates the effectiveness of the shared causal attention layer in improving ADR detection.

4.3.4.3 Case Study

Our proposed SCAN model applies a shared causal feature that allows the model to detect ADRs even in difficult scenarios. Table 4.12 displays a number of example tweets with which the proposed model performed well when compared with the other competing models.

Table 4.9 Results of the ablation study on the WEB_RADR dataset

	Model	Precision	Recall	F1-score
Top 1	Word Only	0.4844	0.9448	0.6404
	Word + POS	0.5194	0.8122	0.6336
	SCAN	0.5094	0.9006	0.6507
Top 5	Word Only	0.5144 ± 0.0376	0.7978 ± 0.1327	0.6168 ± 0.0142
	Word + POS	0.5374 ± 0.0429	0.7425 ± 0.0899	0.6173 ± 0.0095
	SCAN	0.5144 ± 0.0253	0.8420 ± 0.0843	0.6351 ± 0.0112
Top 10	Word Only	0.5282 ± 0.0245	0.7204 ± 0.0845	0.6011 ± 0.0132
	Word + POS	0.5394 ± 0.0263	0.7122 ± 0.0520	0.6090 ± 0.0073
	SCAN	0.5272 ± 0.0171	0.7425 ± 0.0793	0.6102 ± 0.0178

Table 4.10 Results of the ablation study on the combined dataset

	Model	Precision	Recall	F1-score
Top 1	Word Only	0.5639	0.8313	0.6719
	Word + POS	0.5941	0.7460	0.6614
	SCAN	0.5995	0.7922	0.6825
Top 5	Word Only	0.5775 ± 0.0122	0.7808 ± 0.0324	0.6633 ± 0.0056
	Word + POS	0.5920 ± 0.0019	0.7293 ± 0.0142	0.6534 ± 0.0056
	SCAN	0.5879 ± 0.0339	0.7780 ± 0.0587	0.6667 ± 0.0079
Top 10	Word Only	0.5814 ± 0.0115	0.7570 ± 0.0268	0.6566 ± 0.0053
	Word + POS	0.5987 ± 0.0113	0.7066 ± 0.0205	0.6474 ± 0.0049
	SCAN	0.5770 ± 0.0254	0.7840 ± 0.0446	0.6609 ± 0.0055

Tweet 2 in Table 4.12 expresses a direct causal relationship between the drug name and the ADR words *fewer hours sleep*. The SCAN model was able to detect these ADR words, but the other models could not detect them correctly. In our model, we use separate causal features that can capture causal relationships even when the drug name and the ADR words are not in the same sentence in a tweet. For example, in tweets 1 and 4, the drug name and the ADR word are in different sentences, but the SCAN model could still detect the ADR words. We also observed that the SCAN model could detect ADRs even if they appear before the drug name in a sentence. For example, in tweet 3 and 4, the ADRs *jacked up* and *fuzziness* appear before the drug name.

Tweet 5 represents an example of a tweet in which both Cocos et al. and CausalMHA incorrectly labeled the word *poison* as an ADR, but the SCAN model did not. The examples discussed above suggest that our model can generalize well to detect ADR words in various situations.

Table 4.11 Average improvement in the F1 score (%) achieved by the SCAN model against the word only and word + POS models

	Top 1		Top 5		Top 10	
Dataset	**Word Only**	**Word + POS**	**Word Only**	**Word + POS**	**Word Only**	**Word + POS**
ASU_CHOP	-0.41	-0.86	-0.86	-1.28	-1.24	-1.06
SMM4H	1.33	1.41	1.43	1.58	1.33	1.59
WEBADR	1.60	2.70	2.96	2.90	1.51	0.20
Combined	1.57	3.18	0.52	2.03	0.66	2.10

4.3.4.4 Error Analysis

We analyzed the errors made by the proposed SCAN model on the combined dataset, and a few examples of incorrect ADR labeling are displayed in Table 4.13. It was observed that when there were many words between the drug name and ADR words, our model often failed to detect the ADR words, (see tweets 1 and 2 in Table 4.13). In tweet 1, "<user> -nods- My zombie-ness when I first wake up in the morning is mostly from my nightly meds too (<medicine>).", the ADR word is *zombie-ness*, which appears 16 words away from the drug name. Similarly, in tweet 2, "3 day 19 <medicine> diary. just taken 2 paracetamol. probabably will take 70 minutes to work. a long painful night ahead.", the ADR word is *painful*. In this case, the ADR word appear 15 words away from the drug name. Since we use the drug name as a cause of the ADR words, the long distance between a drug name and an ADR word in a tweet may affect the quality of the causal features used to train our model.

Another type of frequent error that was found was incorrectly labeling words related to health conditions as ADRs. Tweet 3, "Fed up of aching bones :(Wonder drugs work some magic please. #rheumatoidarthritis <medicine>", illustrates one such example in which our model incorrectly labeled *aching bones* as an ADR, but in this tweet, it is actually describing a health condition not an ADR. Such examples suggest that the proposed model still has scope for improvement in the future.

Table 4.12 Examples of tweets that were annotated correctly by the SCAN model but not by the other models

IDs	Annotator	Tweets	Labelled ADRs
1	Human Annotator	please tell me that wasn't <medicine>! that stuff is sooo sedating ...	sedating
	Cocos et al.	please tell me that wasn't <medicine>! that stuff is sooo sedating ...	
	CausalMHA	please tell me that wasn't <medicine>! that stuff is sooo sedating ...	
	SCAN	please tell me that wasn't <medicine>! that stuff is sooo sedating ...	sedating
2	Human Annotator	#mhchat <medicine> also causes me to have fewer hours sleep a night	fewer hours sleep
	Cocos et al.	#mhchat <medicine> also causes me to have fewer hours sleep a night	fewer hours
	CausalMHA	#mhchat <medicine> also causes me to have fewer hours sleep a night	to have fewer hours
	SCAN	#mhchat <medicine> also causes me to have fewer hours sleep a night	fewer hours sleep
3	Human Annotator	yeaa bihh all jacked up on <medicine> !	jacked up
	Cocos et al.	yeaa bihh all jacked up on <medicine> !	jacked
	CausalMHA	yeaa bihh all jacked up on <medicine> !	jacked up
	SCAN	yeaa bihh all jacked up on <medicine> !	jacked up
4	Human Annotator	not to worry - just a bit of temporary fuzziness . I've been on <medicine> since 2007 & love it.	fuzziness
	Cocos et al.	not to worry - just a bit of temporary fuzziness . I've been on <medicine> since 2007 & love it.	temporary fuzziness
	CausalMHA	not to worry - just a bit of temporary fuzziness . I've been on <medicine> since 2007 & love it.	temporary fuzziness
	SCAN	not to worry - just a bit of temporary fuzziness . I've been on <medicine> since 2007 & love it.	fuzziness
5	Human Annotator	<medicine> is literally poison. but sometimes you gotta find out the hard way. #headache	
	Cocos et al.	<medicine> is literally poison . but sometimes you gotta find out the hard way. #headache	poison
	CausalMHA	<medicine> is literally poison . but sometimes you gotta find out the hard way. #headache	poison
	SCAN	<medicine> is literally poison. but sometimes you gotta find out the hard way. #headache	

4.4 Summary

In this chapter, we proposed an automatic approach to ADR word detection in tweets by applying a SCAN. We showed that it is possible to improve the word and POS features in the ADR detection task if we combine them causal feature. We also proposed an improved causal feature extraction strategy that overcomes feature overlapping between causal segments, i.e., prefix, midfix, and postfix. In our experiments, we discovered that in our models causal features played a significant role in detecting ADR words in tweets, and the performance of the SCAN model is comparable to or better than the state-of-

Table 4.13 Examples of tweets incorrectly labeled by the SCAN model

IDs	Annotator	Tweets	Labelled ADRs
1	Human Annotator	<user> -nods- My zombie-ness when I first wake up in the morning is mostly from my nightly meds too (<medicine>).	zombie-ness
	SCAN	<user> -nods- My zombie-ness when I first wake up in the morning is mostly from my nightly meds too (<medicine>).	
2	Human Annotator	3 day 19 <medicine> diary. just taken 2 paracetamol. probably will take 70 minutes to work. a long painful night ahead.	painful
	SCAN	3 day 19 <medicine> diary. just taken 2 paracetamol. probably will take 70 minutes to work. a long painful night ahead.	
3	Human Annotator	Fed up of aching bones :(Wonder drugs work some magic please. #rheumatoidarthritis <medicine>	
	SCAN	Fed up of aching bones :(Wonder drugs work some magic please. #rheumatoidarthritis <medicine>	aching bones

the-art models on the publicly available benchmark datasets.

STATEMENT OF CONTRIBUTION TO CO-AUTHORED PUBLISHED PAPER

Chapter 5 includes a co-authored journal paper, which has been submitted to IEEE Transactions on Artificial Intelligence in 2022. The status of the co-authored paper, including all authors, are:

- **Humayun Kayesh**, Md. Saiful Islam, Junhu Wang, "Answering Binary Causal Questions Using Role-oriented Concept Embedding.", IEEE Transactions on Artificial Intelligence (*Submitted*)

My contribution to the paper involved: proposal of deep learning framework and using role-oriented concept embeddings for answering binary casual questions, implementation, experiments, writing and editing manuscript.

(Signed)[redacted](Date) June 10, 2022

Humayun Kayesh

(Countersigned)[redacted](Date) June 10, 2022

Corresponding author of paper: Md. Saiful Islam

(Countersigned)[redacted](Date) June 10, 2022

Supervisor: Junhu Wang

Chapter 5

Answering Binary Causal Questions

Answering binary causal questions is a challenging task, and it requires rich background knowledge to answer such questions. Extracting useful causal features from the background knowledge base and applying them effectively in a model is a crucial step to answering binary causal questions. The state-of-the-art approaches apply deep learning techniques to answer binary causal questions. In these approaches, candidate concepts are often embedded into vectors to model causal relationships among them. However, a concept may play the role of a cause in one question but it could be an effect in another question. This aspect has not been explored extensively in existing approaches. In this chapter, we propose to use role-oriented causal concept embeddings to model causality between concepts. We also propose to use semantic concept similarity to extract causal features from concepts. Finally, we develop a deep learning framework to answer binary causal questions. Our approach achieves comparable or better accuracy than the benchmark approaches.

5.1 Introduction

Binary questions can be answered by yes/no answers. These questions are often asked for the confirmation of given information. Similarly, binary causal questions (BCQs) are

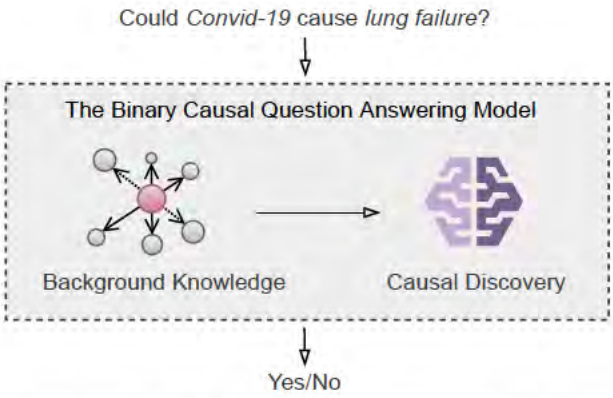


Fig. 5.1 Application scenario of a binary causal question answering model

asked to confirm whether there is a causal relationship between two candidate concepts or not. An example BCQ is illustrated in Fig. 5.1. In the example questions, "Could Covid-19 cause lung failure?", *covid-19* and *lung failure* are two candidate concepts and the question is asking whether there is a causal relationship between them or not.

Extraction and application of background knowledge are some of the key challenges in answering BCQs. A causal knowledge base contains causally related concepts. However, the causal relationships between concepts is directed and concepts play different roles in different contexts. In one context, a concept may play the role of a cause but in another context, the same concept may play the role of an effect. For example, in the following BCQ, "Could accident cause death?", *accident* is a cause concept and *death* is the effect concept. However, in another BCQ, "Could overspeeding cause accidents?", *accidents* is the effect of *overspeeding*. Therefore, a role-oriented approach to causal feature extraction is important for detecting the causal relationship between concepts in a BCQ. Another challenge is to extract useful causal features from the causal knowledge base for each concept in a BCQ. A concept in a BCQ and a concept in the knowledge base need to be matched to extract features. The exact match between knowledge base concepts and input concepts is not always available because of the syntactic diversity of concepts in natural languages.

The recent approaches to BCQ apply transfer learning-based models that are trained on automatically extracted causally-related concept pairs [111, 112]. Hassanzadeh et al. [111] proposed an approach that uses BERT [132] to encode a large

training dataset and then applies a top-k nearest neighbour search technique to answer BCQs. Later, we proposed a transfer learning-based approach [112] that was trained on a smaller dataset while keeping the performance comparable to the previous approach. However, both approaches suffer from low accuracy and precision scores. None of the approaches takes the different roles of the concepts into account. The role-oriented causal concept embedding-based approach proposed in this chapter considers the context-specific roles of the candidate concepts to address this problem. We also address the challenge of causal feature extraction from a causal knowledge base using a semantic concept similarity search technique.

We assume that BCQs contain candidate causal concepts and such questions can be answered by modeling the causal relationship between concepts. Our aim is to model causal relationships between concepts to answer BCQ using a deep learning framework. We achieve this goal by answering the following research questions.

RQ1 *How can we generate the role-oriented causal concept embeddings that can be used in causal discovery between concepts?*

RQ2 *How can we extract rich causal features from causal background knowledge base?*

RQ3 *How can we develop a deep learning framework that utilizes both contextual features and causal features to answer BCQs?*

In *RQ1*, we explore the options to encode the change of roles of a concept in different contexts. In *RQ2*, we study various approaches to extract causal features from the causal knowledge base. The key challenge to answer this question is to map input concepts with the causal knowledge base concepts. In the final research question, *RQ3*, we explore the appropriate structure of a deep learning framework that combines the causal and effect features effectively to model causality between concepts in BCQs. To be specific, our main contributions of this paper are as follows:

- we develop a novel approach to generate role-oriented causal concept embeddings;

- we propose a causal feature extraction approach using semantic concept similarity technique; and
- finally, we develop a novel deep learning framework that combines both contextual and causal features to answer BCQs.

The rest of the paper is organised as follows: our proposed approach to answer BCQs is described in Section 5.2, the experimental settings and results are presented in Section 5.3 and finally, the conclusions are drawn in Section 5.4.

5.2 Our Approach

In this section, we describe our proposed deep learning model to answer BCQs. The proposed framework consists of three main components: causal feature extraction, contextual feature extraction, and knowledge fusion. Below, we describe each component in detail. The architecture of the proposed framework is illustrated in Fig. 5.2.

5.2.1 Causal Features Extraction

The causal feature extraction module extracts causal features from the background knowledge base.

5.2.1.1 Causal Concept Network

As the first step to causal features extraction, we build a causal concept network using a high-quality causal knowledge base. We use CauseNet[1], which is a dataset of causally related concepts published by Heindorf et al. [133], as our causal knowledge base. The authors published two versions of their dataset: CausaNet-full and CauseNet-precision.

[1]https://causenet.org

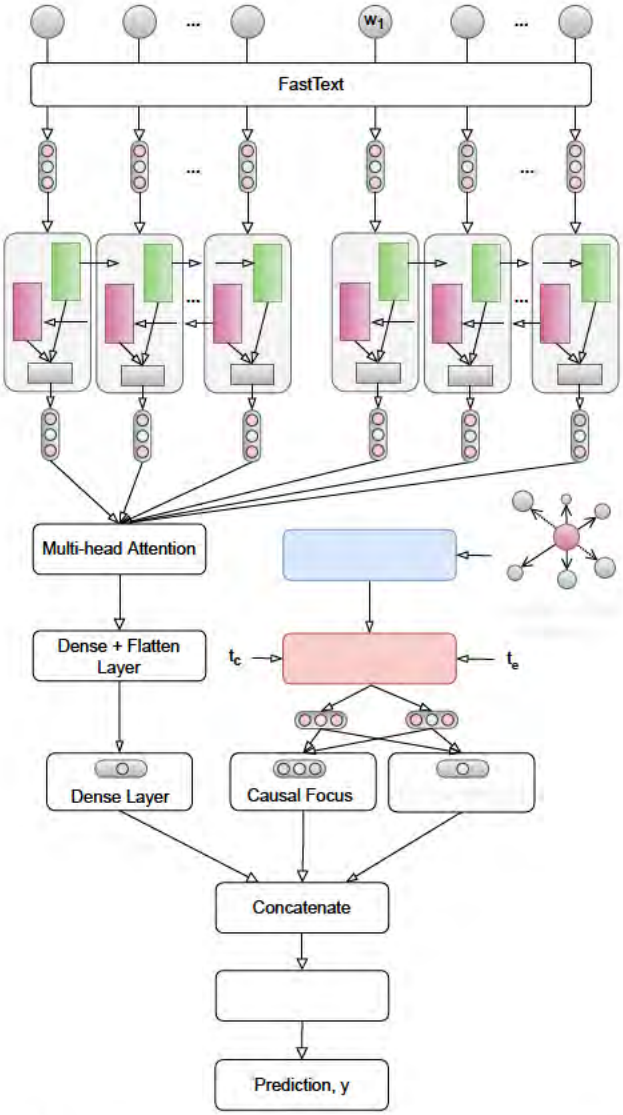


Fig. 5.2 The proposed deep learning framework for answering binary causal questions

We use the latter one as it contains high precision causal concepts. The dataset consists of more than 197K concept pairs and around 80k unique concepts. We use this dataset to prepare our causal concept network, G, where each node is a concept and each edge represents a causal relationship. After preparing G, we generate embeddings for each node, v, in G. A concept may be a cause for another concept, it could also be an effect of another concept, i.e., there are two possible roles of each concept. Therefore, we need two separate embeddings to represent each role effectively.

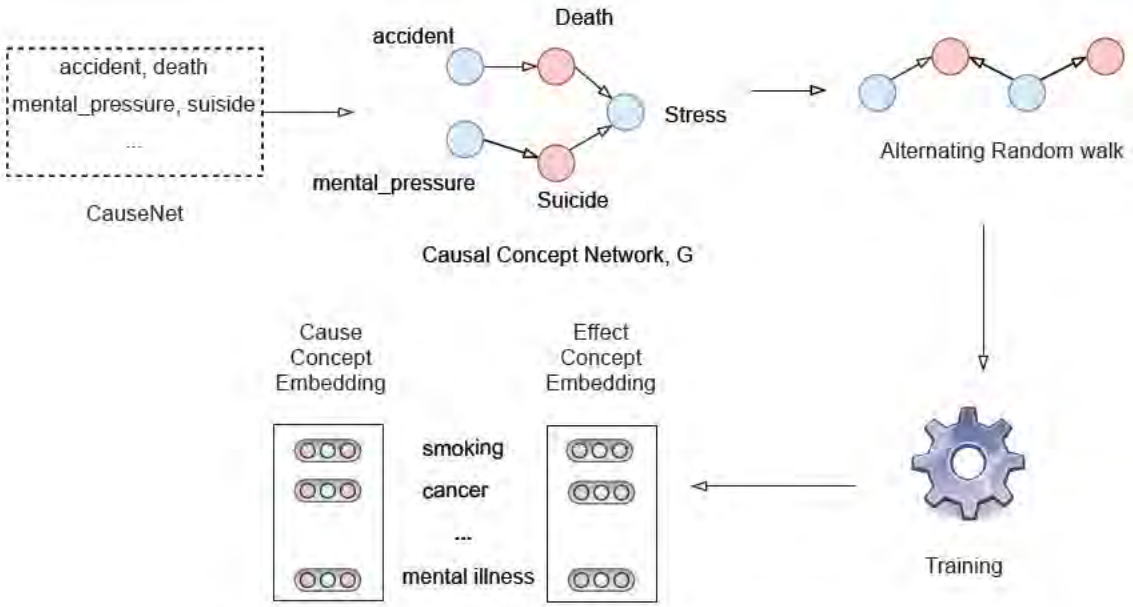


Fig. 5.3 Role-oriented concept embedding generation

5.2.1.2 Role-oriented Concept Embedding

We propose to use the alternating random walk technique [134] to generate the role-oriented embeddings of each node in G. The technique was originally proposed to generate embeddings of each node of a directed graph based on their roles. The role is determined by the inward or outward links in the directed graph. In our case, we have causal concepts as nodes and directed edges representing the orientation of the causal relationship. Fig. 5.3 illustrates the technique to generate role-oriented concept embeddings from G. An inward link to a node *death* from *accident* in G represents that *death* is an effect of *accident*. Similarity, an outward link from a node *death* to *stress* represents that *death* is a cause of *stress*.

As a first step to generate concept embedding, a bipartile undirected graph, G', is generated from G. It is ensured that the adjacency matrix of G' is symmetric. Then, input nodes are sampled from G and an alternating walk is performed from each input node. To preserves the roles of neighborhood nodes, the walk generates sample paths where odd nodes are the cause concepts and even nodes are the effect concepts. The alternating walk is the combination of two types of walks: the source walk and the target walk. The source walk starts from a cause node and each alternating node in the path is a

cause node. Similarly, the target walk starts for an effect node and each alternating node is an effect node. Finally, the model is jointly trained on the generated random walks and it generates two role-oriented embeddings for each node, v. For example, the *accident* and *death* nodes will have two embeddings each: cause and effect embeddings. Because *accident* and *death* are causally-related in G, the causal embedding of the *accident* and the effect embedding of *death* should be closer in the embedding space. We refer the reader to [134] for a detailed description of the alternating random walk technique.

5.2.1.3 Semantically Similar Concepts Discovery

In this step, we extract role-oriented concept embedding for an input concept pair, (t_c, t_e), from the cause and effect embeddings generated in the previous step. Here, t_c is a cause concept and t_e is an effect concept. Each concept may contain one or more words. The key challenge in this task is to map between an input concept and the concept nodes in G. The most obvious approach would be to apply a hit and miss scenario. If a concept matches word-to-word with a node in G, this is a hit and the corresponding embedding is returned. Otherwise, a default embedding is returned. However, due to the dynamic nature of natural languages, two semantically similar concepts can be written in many different forms. To address this challenge, we propose to use a semantic concept similarity technique to map input concepts to the those in G.

To find semantically similar concepts, first, we convert the input concepts, t_c and t_e, into vectors using a pretrained universal sentence encoder [135] model. We also encode each concept in G using the same universal sentence encoder model that is used for t_c and t_e. Then, we calculate cosine similarity between the vector of t_c and each encoded nodes in G to find the most similar concepts. Similarly, we find the semantically similar concepts in G for t_e. Then, we extract the corresponding role-oriented concept embeddings for each concept from the embeddings generated in the previous step. We denote the concept embeddings of t_c and t_e as v_c and v_e, respectively. Fig. 5.4 illustrates the process of semantically similar concepts discovery for an example candidate cause and effect concept pair.

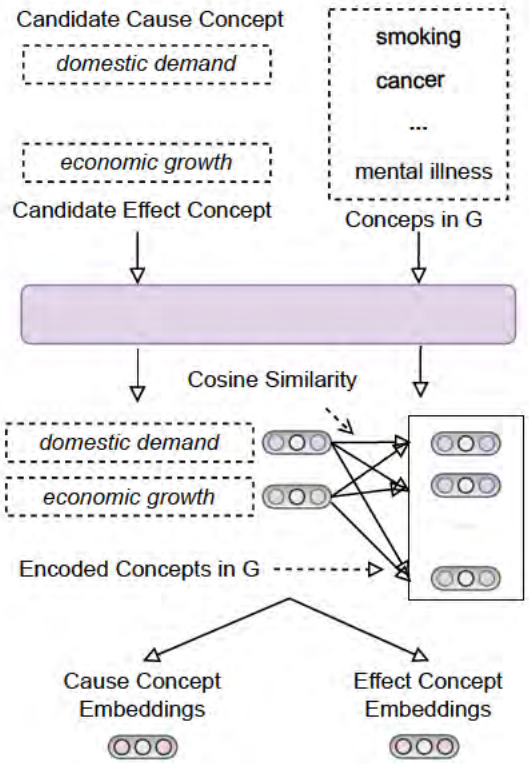


Fig. 5.4 Semantically similar concepts discovery

5.2.2 Contextual Features Extraction

The inputs to this module are the same cause-effect concept pair (t_c, t_e) as the previous module. The contextual features are extracted from the concepts using a BLSTM [56] and a multi-head attention technique. First, we use a pretrained fastText model [136] to convert the input concepts into embeddings. Then, we apply a BLSTM on the cause concept embeddings and another BLSTM on the effect concept embeddings. The outputs of BLSTM layers are passed to a multi-head attention layer followed by a series of hidden layers to capture the contextual features. We describe each step of the proposed contextual feature extraction below.

- **Tokenization** - A concept may contain one or more words. In this step, t_c and t_e are passed to a pretrained fastText tokenizer that splits each concept into tokens. The tokenizer also contains a vocabulary D which is a dictionary of tokens and their indices. The tokenizer then converts each sequence of tokens into a sequence of indices. We denote the sequence of t_c as x_c and t_e as x_e. To avoid variable-length

inputs, the tokenizer applies padding and truncating techniques on the sequences.

- **Embedding layer** - In this step, we build embedding matrices, $R^{z \times d}$, from cause-effect sequences, using the pretrained tokenizer model, where z is the number of tokens in D and d is the embedding dimension. Each row in R corresponds to the token in the same index in D. we convert x_c and x_e into matrices by replacing each token index with its corresponding embedding in D. We denote the corresponding embedding matrices of x_c and x_e as $X_c^{l \times d}$ and $X_e^{l \times d}$, respectively, where l is the maximum length of sequence.

- **BLSTM layer** - The BLSTM layer in the proposed model is responsible for extracting context information from each concept. Following Kayesh et al. [137], we apply two separate BLSTM models on X_c and X_e. A BLSTM is consists of a forward LSTM and a backward LSTM. As shown in Eq. 5.1, the output of forward LSTM, $\overrightarrow{h_c}$, and backward LSTM, $\overleftarrow{h_c}$, are combined to prepare cause contextual features, h_c, where $\oplus$ denotes a concatenation operation. Similarly, the effect contextual features, h_e, are extracted as shown in Eq. 5.2.

$$h_c = \overrightarrow{h_c} \oplus \overleftarrow{h_c} \tag{5.1}$$

$$h_e = \overrightarrow{h_e} \oplus \overleftarrow{h_e} \tag{5.2}$$

- **Attention layer** - We pass the cause and effect concept features to a multi-head attention layer to extract the attention weights between the cause and effect contextual features. As shown in Eq. 5.3, h_c and h_e are used as the inputs to the attention layer. The output of the layer is denoted as h and f_a is a multi-head attention function proposed by Vaswani et al. [67].

$$h = f_a(h_c, h_e) \tag{5.3}$$

- **Dense layer** - In this step, the output of the attention layer, h, is passed to a dense layer to reduces its dimension. Each neuron in this layer applies a ReLU activation function on the input as shown in Eq. 5.3 where W is a weight matrix b is the bias

matrix.

$$h' = ReLU(h \cdot W + b) \tag{5.4}$$

- **Flatten layer** - In this layer, we flatten the dense layer output, h', and prepare a single vector, v_m. This vector is then passed to another dense layer to generate a single value contextual feature, l, extracted from candidate cause and effect concepts. Eq. 5.5 shows the calculation of l where w and b_m are the weights and biases.

$$l = ReLU(v_m \cdot w + b_m) \tag{5.5}$$

5.2.3 Knowledge Fusion

In this section, we describe our proposed technique to combine contextual and causal features to develop a deep learning model to answer BCQs. Designing an appropriate structure of the model that combines two features effectively is a challenging task. The model needs to be aware that the distance between concept embeddings for candidate cause and effect concepts are crucial for answering a BCQ. If a candidate cause is closer to the candidate effect in the embedding space, the causal features extracted from G should get higher priority. To this end, we propose a two-way approach to add the extracted causal features to the model.

5.2.3.1 Causal Focus (CF)

The causal features extraction step extracts v_c and v_e from t_c and t_e, respectively. To encode the interplay between cause and effect in the embedding space, we extract the causal focus features. To prepare causal focus features, r_f, we perform an element-wise multiplication between v_c and v_e as shown in Eq. 5.6. We find in our experiments that causal focus is an effective feature in modeling causal relationships between concepts.

$$r_f = (v_c \times v_e) \tag{5.6}$$

5.2.3.2 Causal Strength (CS)

The dot product between concept embeddings, v_c and v_e, is an indicator of causal strength between the input concepts, t_c and t_e. To model this causal feature we perform a dot product of v_c and v_e and we denote the output as r_s.

After preparing the contextual feature l and the causal features r_f and r_s, we concatenate them together into a single feature vector. We allow the model to learn the weights, W_y, and biases, b_y, from the features by passing it to a dense layer that uses a Sigmoid function as the activation function. Eq. 5.7 shows the steps where y is the final predicted output.

$$y = \sigma((l \oplus r_f \oplus r_s) \cdot W_y + b_y) \tag{5.7}$$

5.3 Experimental Evaluation and Analysis

In this section, we discuss our training and evaluation datasets, benchmark models, and experimental settings. Then, we present our experimental results and compare them with the benchmark models. Finally, future research challenges have been discussed to advance this field.

5.3.1 Database Setup

We trained our model on an automatically generated database that was extracted from 1 million news articles [138]. We applied the dataset preparation technique described in Section 5.3.2.2 and we refer this dataset as 'News Articles' in our experiments. The

Table 5.1 Examples from the training and evaluation datasets

Dataset	Cause	Effect	Label
News Articles	poisonous words	homes and marriages have been destroyed	causal
	the last time jim harbaugh coached a football	game showed he why he is so coveted	not causal
CauseNet	global warming	extinction	causal
	fascism	poor diet	not causal
CE Pairs	broadband access	more new businesses	causal
	increased growth	dent consumer and business confidence	not causal
NATO-SFA	consistent	Vulnerabilities	causal
	climate change	new opportunities	not causal
Risk Models	growing social tension	reduced tourism	causal
	rising regional tension	resource competition	not causal
SemEval	collision	fire	causal
	protein	researchers	not causal
Twitter	families truly suport girl-child	we can se that sky to is not the limit	causal
	i ned to be front and centre	it's al about me	not causal

dataset contains 100K positive examples and an equal number of negative examples. We also trained our model on the CauseNet dataset. In this case, we automatically prepared an equal number of negative examples from the same dataset by applying the following approach.

- First, we randomly sampled the cause and effect concepts (with replacement) but swapped their positions. We used the effect concept samples as the causes and the cause concept samples as the effects in the negative examples. This technique reduces the chance of an actual causal concept pair being added to the negative examples.

- Then, we applied further filtering to remove the concept pairs that have cosine similarity scores higher than 0.2. The sampling size was twice the size of the positive concept pairs. After discarding the pairs with higher cosine similarities, we kept the size of the negative and positive examples equal. Here, we used the

same role-oriented concept embeddings as described in Section 5.2.1 for the cosine similarity calculation.

We evaluated all models on the same evaluation datasets used in our previous work [112]. The evaluation datasets used in our experiments are as follows:

- **Risk Models** - This dataset was prepared from a set of decision support system models [139, 140] that represent event-event relationships as graphs. A node represents an event and an edge represents a cause-effect relationship. This dataset contains 804 cause-effect pairs.
- **CE Pairs** - This dataset is an extension of the Risk Models dataset. Seven human annotators were used to find the causes and effects of a subset of nodes in the models using a web search. The nodes and the corresponding search results were used as the cause-effect pairs. There are 302 such pairs in the dataset.
- **NATO-SFA** - This dataset was extracted from a report of Strategic Foresight Analysis (SFA) [141] published by the North Atlantic Treaty Organization (NATO) in 2017. The human-generated report contains the changes to the world (causes) and their implications (effects). The dataset contains 118 such concept pairs and their labels.
- **SemEval** - This dataset is collected from SemEval 2010 sub-task 8 [142] and the size of the dataset is 1730. Each example contains a pair of labeled concepts, e.g., (collision, fire) is a pair of concepts and labeled as causal.
- **Twitter** - This dataset was prepared by Kayesh et al. [137, 143, 144] by capturing tweets related to Commonwealth Games 2018. The cause-effect pairs were extracted from Tweets using causal cue words. The dataset contains 916 pairs of causes and effects.

In each dataset, 50% of the pairs are causal and the remaining 50% are not_causal. Table 5.1 presents one causal and one not_causal pair from each dataset used in this experiment.

5.3.2 Benchmark Models

5.3.2.1 Existing Models in the Literature

In this section, we describe the existing models in the literature that we used in this experiment as benchmark models and compared our proposed models with them.

- **PMI** - The PMI score is the co-occurrence strength between two words. Hassanzadeh et al. [111] extended this model of calculating PMI scores between two text spans. The model converts the candidate cause-effects concepts into two bags of phrases and then calculates the average PMI scores between all phrase pair combinations.
- **CEA** - The cause-effect association (CEA) model was proposed by Do et al. [26] that combines PMI score with joint inverse document frequency score to calculate the causal strength between a cause-effect event pair.
- **DCC** - This model was proposed by Hassanzadeh et al. [111]. The model indexes the list of cause-effect pairs and looks for an exact match for a candidate cause-effect pair. The model calculates the causal strength between the cause-effect pair by calculating its number of hits.
- **DCC-embed** - This model trains a customized version of word2vec model to generate phrase embeddings [111]. Then, it performs a nearest neighbour search to find the top-k closest cause and effect phrases. The search results are then used to calculate the causal relationship between the cause-effect pair.
- **NLM-BERT-17M** - This neural network model was proposed by Hassanzadeh et al. [111]. The model applies a BERT model to encode 17M causal sentences. Given a causal-effect concept pair (X, Y), the model converts it into two sentences: "X may cause Y" and "Y may cause X". The top-k similar sentences are then searched for each sentences and the average cosine similarity scores are calculated. Two threshold values, which are specific to each test dataset, are then applied

to the cosine similarity values to decide the causal relationship between concepts. Because we have used the same evaluation datasets (except for the Twitter dataset) and their training dataset is not publicly accessible, we reported the results of the above-mentioned models directly from their paper.

- **NLM-BERT** - This model was proposed by Hassanzadeh et al. [111] but we implemented this model ourselves to train it on our smaller training dataset and compared it with our proposed models. In the evaluation, we used the same test dataset dependent thresholds used by the authors for each test dataset.
- **NLM-BERT++** - This model is the same as NLM-BERT, but instead of selecting different thresholds for different test datasets, we used a single pair of thresholds for all test datasets. We used the automatic threshold selection technique proposed by Kayesh et al. [112]. This technique empirically discovers the optimal thresholds from the training datasets and does not depend on the test dataset.
- **CausalNet** [7]- This approach builds a causal network from training data and calculates causal scores using the network. In this experiment, we used our News Articles training data to build the causal network. In our evaluation, we consider a given candidate concept pair as causal if the causal score is greater than zero. Otherwise, we consider the pair as not_causal.

5.3.2.2 BERT-based Models

In our previous work [112], we proposed a transfer learning-based approach for answering BCQ. As a part of that work, we prepared a training dataset that were extracted from news articles. We used a set of causal cue words, e.g. because, as a result of, and due to, to extract causal pairs from articles. The following example "heavy rain *causes* traffic jams" contains two causally related concepts, *heavy rain* and *traffic jams*. The full list of causal cue words used to build the dataset can be found in [112]. We extracted such concept pairs and labeled them as *causal*. To extract negative examples, we collected the sentences that contains no causal cue words and divided them into halves to prepare the negative pairs. These pairs were then labeled as *not_causal*.

We fine-tuned pretrained BERT-based models on the training dataset mentioned-above to answer BCQs. Pretrained models, such as BERT and RoBERTa [145], are rich in linguistic knowledge as they are trained on large datasets. Before using the training data for fine-tuning, we prepared full sentences from the concept pairs. For example, a concept pair (X, Y) is presented as "X may cause Y". Such sequences are then passed to the BERT tokenizer that tokenizes the sequences and prepares them as per models accepted input format. The preprocessed sequences are then used to fine-tune the transfer learning models and the fine-tuned models were then saved for prediction and evaluation. In this chapter, we use these models as the benchmark models to show the effectiveness of the proposed approach.

We implemented the following BERT-based models in [112] following the technique described above to answer BCQs:

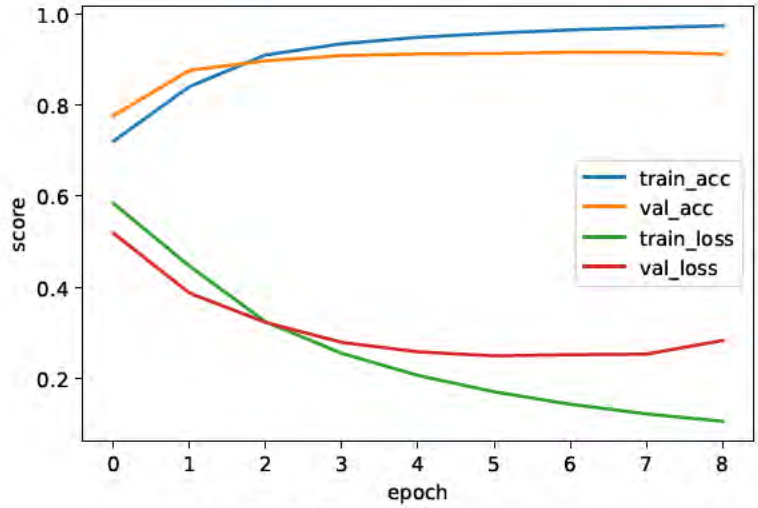
- **BERT** [132] - BERT is a transfer learning based model that was achieved state-of-the-art results in many tasks including question answering. In this experiment, we used a pretrained BERT model provided by the Huggingface[2] transformers library [146]. We used the 'bert-base-uncased' version of the pretrained model.

- **RoBERTa** [145] - RoBERTa is a variation of BERT that optimizes training and hyperparameter tuning of the original model and achieves state-of-the-art results in text classification and question answering tasks. We used the 'roberta-base' version of the pretrained model from Huggingface.

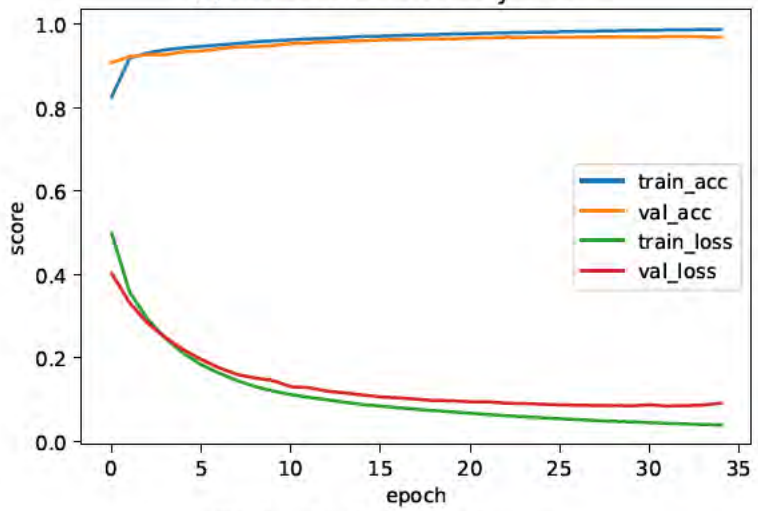
- **DistilBERT** [147] - The authors of this model reduced the structural complexity and optimized the hyperparameter tuning of the original BERT model. The objective of this model was to develop a lightweight version of BERT without sacrificing the performance too much. We used the 'distilbert-base-uncased' version of the pretrained model from Huggingface.

- **ALBERT** [148] - ALBERT is also a variation of BERT and it was proposed focusing on scalability. The model optimizes training time and memory usage for large

[2]https://github.com/huggingface/transformers



(a) On the News Articles dataset



(b) On the CauseNet dataset

Fig. 5.5 Training history of our proposed model (Causal Focus + Causal Strength + Contextual Features)

datasets. In our experiments, we used the 'albert-base-v2' version of the pretrained model from Huggingface.

5.3.3 Experiment Settings

We split our training datasets into training, validation, and test sets. We used 80% of the training dataset for training and the remaining 20% of the data were split into halves for validation and testing. We evaluated our model on five benchmark datasets as described

in Section 5.3.1. We implemented our models in Python and used TensorFlow[3] and Keras[4] to construct deep learning structures. We used a pretrained fastText embedding model[5] to train our model. The fastText model was trained on the Common Crawl[6] and Wikipedia[7] data and contains 300 dimension vectors. For semantic similarity calculation, we used a retrained universal sentence encoder model[8] that returns a 512 dimension vector for the given text.

We trained our model for 300 epochs while setting the patience parameter to 3. To prepare the role-oriented concept embeddings, we followed the instructions provided in Khosla et al. [134] and their code repository[9]. The concept embeddings generation model was trained to produce two 300 dimension vectors per concept in G, a causal embedding, and an effect embedding. We set the parameters walkSize to 1, samples to 100, rho to 0.025, and joint to 0. We also set the parameter negative to 0 because there was no negative pairs in CauseNet. Rather, we trained the role-oriented causal concept embedding model on the randomly selected negative pairs. We ran our experiments on a machine with 16 core Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz and 128GB of RAM.

Since the calculation of cosine similarity between two datasets is a computationally expensive operation, we pre-extracted top-10 semantically similar concepts from G for each cause and effect concept in News Articles training dataset and all the evaluation datasets. However, in this experiment, we used only the top-1 concept embeddings. When training with the CauseNet dataset, we did not use the semantic similarity calculation as the concepts in CauseNet already exist in G. Therefore, we used the exact matching technique to extract cause and effect concept embeddings for the CauseNet dataset.

[3]https://www.tensorflow.org/
[4]https://keras.io/
[5]https://fasttext.cc/docs/en/crawl-vectors.html
[6]https://commoncrawl.org/
[7]https://www.wikipedia.org/
[8]https://tfhub.dev/google/universal-sentence-encoder/4
[9]https://git.l3s.uni-hannover.de/khosla/nerd

Table 5.2 Evaluation of the proposed deep learning model(s) on the test sets of the training datasets

Training Data	**Models**	**# Pairs**	**Acc**	**Pre**	**Rec**	**F1**
News Articles	CF	100K	0.5100	0.5100	**1.0000**	0.6700
	CS	100K	0.5000	0.8700	0.0000	0.0100
	Context	100K	0.8700	0.8200	0.9600	0.8800
	CF + CS	100K	0.5100	0.5200	0.2500	0.3400
	CF + Context	100K	0.9200	0.9200	0.9100	0.9200
	CF + CS + Context	100K	0.9100	0.9200	0.9100	0.9100
CauseNet	CF	197K	0.6848	0.9597	0.3930	0.5577
	CS	197K	0.7663	**1.0000**	0.5376	0.6993
	Context	197K	0.9351	0.9611	0.9084	0.9340
	CF + CS	197K	0.9483	0.9650	0.9315	0.9479
	CF + Context	197K	0.9522	0.9485	0.9574	0.9529
	CF + CS + Context	197K	**0.9698**	0.9648	**0.9758**	**0.9703**

5.3.4 Results and Discussion

We separately trained our proposed model on the News Articles and CauseNet datasets and we tested the models on the test set. The test set consists of 10% data from the corresponding training dataset. Fig. 5.5 shows the training history of our proposed model and Table 5.2 displays the performance of the different variations of our proposed model on the test sets of News Articles and CauseNet datasets. We denote Causal Focus, Causal Strength, and Contextual features as CF, CS, and Context, respectively, when presenting the evaluation results. Our proposed approach performed better on the CauseNet dataset compared to the News Articles dataset when validated on the test sets of training datasets. The results in Table 5.2 suggest that in terms of accuracy and f1-score, our models perform comparatively better when trained on the CauseNet dataset. Also, the *CF + CS + Context* model, which uses causal focus, causal strength, and contextual features, achieved the highest accuracy of 0.9698 and f1-score of 0.9703.

Table 5.3 to 5.7 presents our experimental results on the evaluation datasets. Our benchmark paper, [111], reported two types of results per model: maximum f1-score and accuracy. When comparing our models with the benchmark models proposed in [111], we compared against the models with maximum accuracy values as our aim is to improve

Table 5.3 Evaluation results on the CE Pairs dataset

Training Data	**Models**	**# Pairs**	**Acc**	**Pre**	**Rec**	**F1**
-	PMI	-	0.5090	0.6360	0.0440	0.0820
-	CEA	-	0.5410	0.6100	0.2250	0.3290
-	DCC	-	0.5590	0.6070	0.3380	0.4340
-	DCC-embed	-	**0.5750**	0.6110	0.4130	0.4930
-	NLM-BERT-17M	17M	0.5630	0.6470	0.2750	0.3860
News Articles	NLM-BERT	100K	0.5000	-	-	-
	NLM-BERT++	100K	0.5000	0.5000	1.0000	0.6667
	CausalNet	100K	0.5188	0.5156	0.6188	0.5625
	BERT	100K	0.5062	0.5032	0.9688	0.6624
	RoBERTa	100K	0.4750	0.4870	0.9375	0.6410
	DistilBERT	100K	0.5125	0.5064	0.9875	**0.6695**
	ALBERT	100K	0.4750	0.4868	0.9187	0.6364
	CF	100K	0.5000	0.5000	1.0000	0.6667
	CS	100K	0.5000	-	-	-
	Context	100K	0.4594	0.4764	0.8188	0.6023
	CF + CS	100K	0.5094	0.5185	0.2625	0.3485
	CF + Context	100K	0.4781	0.4868	0.8063	0.6071
	CF + CS + Context	100K	0.4438	0.4640	0.7250	0.5659
CauseNet	CF	197K	0.5000	-	-	-
	CS	197K	0.5000	-	-	-
	Context	197K	0.5375	0.5291	0.6813	0.5956
	CF + CS	197K	0.5094	1.0000	0.0188	0.0368
	CF + Context	197K	0.5344	0.5258	0.7000	0.6005
	CF + CS + Context	197K	**0.5750**	0.5723	0.5938	**0.5828**

model accuracy while keeping a balanced precision and recall.

Table 5.3 presents the evaluation results on the CE Pairs dataset. The results suggest that the proposed model, *CF + CS + Context*, achieved the same maximum accuracy score of 0.5750 as the benchmark model DCC-embed. However, our model achieved more balanced precision, 0.5723, and recall, 0.5938 than DCC-embed. The precision of DCC-embed was better than *CF + CS + Context* but recall was below 0.50. Another benchmark, DistilBERT, achieved the highest f1-score of 0.6695 but its accuracy score of 0.5125 was comparatively low. Tables 5.4 and 5.5 displays the evaluation results on the NATO-SFA and Risk Models datasets. Our proposed approach suffers on the NATO-SFA datasets in terms of accuracy. On the NATO-SFA dataset, the benchmark model

Table 5.4 Evaluation results on the NATO-SFA dataset

Training Data	**Models**	**# Pairs**	**Acc**	**Pre**	**Rec**	**F1**
-	PMI	-	0.6020	0.6760	0.3900	0.4950
-	CEA	-	0.5510	0.6500	0.2200	0.3290
-	DCC	-	0.6610	0.7020	0.5590	0.6230
-	DCC-embed	-	**0.6690**	0.7630	0.4920	0.5980
-	NLM-BERT-17M	17M	0.5590	0.5380	0.8470	0.6580
News Articles	NLM-BERT	100K	0.5000	-	-	-
	NLM-BERT++	100K	0.5000	0.5000	1.0000	**0.6667**
	CausalNet	100K	0.5339	0.5556	0.3390	0.4211
	BERT	100K	0.5000	0.5000	0.9322	0.6509
	RoBERTa	100K	0.4915	0.4956	0.9492	0.6512
	DistilBERT	100K	0.5000	0.5000	0.9661	0.6590
	ALBERT	100K	0.4661	0.4818	0.8983	0.6272
	CF	100K	0.5000	0.5000	1.0000	0.6667
	CS	100K	0.5085	1.0000	0.0169	0.0333
	Context	100K	0.5085	0.5049	0.8814	0.6420
	CF + CS	100K	0.4746	0.4681	0.3729	0.4151
	CF + Context	100K	0.5000	0.5000	0.8983	0.6424
	CF + CS + Context	100K	0.5085	0.5048	0.8983	0.6463
CauseNet	CF	197K	0.5000	-	-	-
	CS	197K	0.5085	1.0000	0.0169	0.0333
	Context	197K	0.4746	0.4851	0.8305	0.6125
	CF + CS	197K	0.5339	1.0000	0.0678	0.1270
	CF + Context	197K	0.5000	0.5000	0.7966	0.6144
	CF + CS + Context	197K	0.5508	0.5429	0.6441	0.5891

DCC-embed achieved the highest accuracy of 0.6690 but the model's recall was below 0.5. Our best model on this dataset was *CF + CS + Context* that achieved a lower accuracy and recall than the benchmark model. However, the model's recall of 0.6441 was comparatively better and the f1-score was comparable to DCC-embed. On the Risk Models dataset, the accuracies of both benchmark models and proposed models were low. The highest accuracy score of 0.5570 on this dataset was achieved by the transfer learning-based model NLM-BERT-17M that was trained on a dataset of 17 million causal pairs. However, the proposed *CF + CS + Context* model achieved a comparable accuracy and f1-scores although the model uses a smaller training dataset of 197K causal concept pairs.

Table 5.6 presents the evaluation results on the SemEval dataset. On this dataset, the proposed model *CF + CS* achieved the best accuracy of 0.7543 with a balanced precision and recall of 0.7397 and 0.7850, respectively. The SemEval dataset contains causal concept pairs without any contextual information. This is why the models with causal features only outperformed the models with contextual features on this dataset. The benchmark model DCC-embed achieved the closest accuracy of 0.7340 compared to *CF + CS* but the recall score of this benchmark model is only 0.5790. The results on table 5.7 suggests that the proposed *CF + Context* model achieved the best accuracy on the Twitter dataset. Another model *CF + CS + Context* achieved a comparable accuracy of 0.5764. The benchmark model with the closest accuracy of 0.5426 was achieved by the CausalNet model but recall of this model was close to 0.5.

Overall, our proposed approach achieved comparable or better accuracy over the benchmark models on all the datasets except for NATO-SFA. The proposed models that were trained on the CauseNet dataset achieved comparatively higher accuracy on the evaluation datasets (except for SemEval) compared to the proposed models trained on the news articles dataset. This result suggests that a dataset with high quality concept pairs can improve accuracy. We found that CF and CS plays an important role in causality detection between concept pairs. They were present in all of our best performing models on the evaluation datasets, which answers our *RQ1* and *RQ2*. We also found that the models with only the causal features (CF + CF) achieved higher precision but lower recall, which resulted in lower accuracy and f1-scores. The proposed fusion technique that combines the contextual and causal features achieved a better accuracy among the proposed models for all datasets except for the SemEval dataset. These results suggest that our deep learning framework is effective in answering BCQs, which answers our *RQ3*.

5.3.5 Future Research Challenges

In this chapter, our proposed approach achieved better accuracy and more balanced precision and recall compared to the benchmark models. However, there are some remaining challenges that need to be addressed to advance the field.

Table 5.5 Evaluation results on the Risk Models dataset

Training Data	Models	# Pairs	Acc	Pre	Rec	F1
-	PMI	-	0.5310	0.6190	0.1630	0.2580
-	CEA	-	0.5430	0.5470	0.5030	0.5240
-	DCC	-	0.5080	0.5340	0.1280	0.2060
-	DCC-embed	-	0.5200	0.5630	0.1820	0.2750
-	NLM-BERT-17M	17M	**0.5570**	0.5510	0.6140	**0.5810**
News Articles	NLM-BERT	100K	0.5000	0.5000	1.0000	0.6667
	NLM-BERT++	100K	0.5000	0.5000	1.0000	0.6667
	CausalNet	100K	0.4900	0.4938	0.7960	0.6095
	BERT	100K	0.5025	0.5013	0.9900	0.6656
	RoBERTa	100K	0.4975	0.4987	0.9701	0.6588
	DistilBERT	100K	0.4950	0.4974	0.9677	0.6571
	ALBERT	100K	0.4988	0.4994	0.9627	0.6576
	CF	100K	0.5000	0.5000	1.0000	0.6667
	CS	100K	0.5012	1.0000	0.0025	0.0050
	Context	100K	0.4677	0.4819	0.8607	0.6179
	CF + CS	100K	0.4950	0.4868	0.1841	0.2671
	CF + Context	100K	0.4851	0.4914	0.8557	0.6243
	CF + CS + Context	100K	0.4726	0.4838	0.8184	0.6081
CauseNet	CF	197K	0.5000	-	-	-
	CS	197K	0.5012	1.0000	0.0025	0.0050
	Context	197K	0.4938	0.4959	0.7438	0.5950
	CF + CS	197K	0.5025	0.7500	0.0075	0.0148
	CF + Context	197K	0.5348	0.5292	0.6318	0.5760
	CF + CS + Context	197K	**0.5323**	0.5297	0.5771	**0.5524**

- **High precision, low recall** - The benchmark models including ours exhibited imbalanced precision and recall in certain cases. The low recall is the result of two concepts that are causality-related being far distant in the embedding space. Experiments suggest that a dataset with high-quality concept pairs leads to better performance, e.g., results on the SemEval dataset as shown in Table 5.6. However, the development of a high-quality dataset of causally-related concepts to answer BCQs requires a significant effort. Since a manual approach is time-consuming (e.g., CauseNet), an automatic extraction of high-quality causal concept pairs with performance guarantee can be explored in the future to address this issue.

- **Concept mapping** - Our training and evaluation datasets are from different

Table 5.6 Evaluation results on the SemEval dataset

Training Data	Models	# Pairs	Acc	Pre	Rec	F1
-	PMI	-	0.5290	0.5300	0.5110	0.5200
-	CEA	-	0.5400	0.5420	0.5260	0.5340
-	DCC	-	0.7200	0.7930	0.5970	0.6810
-	DCC-embed	-	0.7340	0.8380	0.5790	0.6850
-	NLM-BERT-17M	17M	0.6190	0.6260	0.5930	0.6090
News Articles	NLM-BERT	100K	0.5936	0.7812	0.2601	0.3903
	NLM-BERT++	100K	0.5029	0.5014	1.0000	0.6680
	CausalNet	100K	0.5168	0.6124	0.0913	0.1590
	BERT	100K	0.5266	0.5137	0.9965	0.6779
	RoBERTa	100K	0.5006	0.5003	1.0000	0.6669
	DistilBERT	100K	0.5052	0.5026	0.9988	0.6687
	ALBERT	100K	0.5254	0.5132	0.9873	0.6754
	CF	100K	0.5000	0.5000	1.0000	0.6667
	CS	100K	0.5977	0.9721	0.2012	0.3333
	Context	100K	0.5040	0.5020	0.9977	0.6680
	CF + CS	100K	**0.7543**	0.7397	0.7850	**0.7616**
	CF + Context	100K	0.5029	0.5014	1.0000	0.6680
	CF + CS + Context	100K	0.5017	0.5009	1.0000	0.6674
CauseNet	CF	197K	0.6474	0.9923	0.2971	0.4573
	CS	197K	0.5977	0.9721	0.2012	0.3333
	Context	197K	0.4532	0.4700	0.7341	0.5731
	CF + CS	197K	0.7173	0.9372	0.4659	0.6224
	CF + Context	197K	0.5069	0.5048	0.7353	0.5986
	CF + CS + Context	197K	0.5671	0.5507	0.7283	0.6272

sources, which is why it was challenging to train to map an unknown concept to one of its training concepts. We have partly addressed this problem by discovering semantically similar concepts from the CauseNet for an input concept. We have applied universal sentence encoder and cosine similarity to find the most similar concepts by comparing a given concept to all the concepts in the CauseNet. However, this approach does not always guarantee to return the causally related concepts. A better language-based modeling technique can be explored to map the training concepts to the concepts in the test datasets with performance guarantee.

- **Quality of embeddings** - We trained our role-oriented causal concept embedding model on the positive edge samples extracted from the concept

Table 5.7 Evaluation results on the Twitter dataset

Training Data	Models	# Pairs	Acc	Pre	Rec	F1
	NLM-BERT-17M	17M	-	-	-	-
News Articles	NLM-BERT	100K	0.4989	-	-	-
	NLM-BERT++	100K	0.5011	0.5011	1.0000	0.6676
	CausalNet	100K	0.5426	0.5472	0.5054	0.5255
	BERT	100K	0.5044	0.5028	0.9673	0.6617
	RoBERTa	100K	0.5131	0.5073	0.9826	0.6691
	DistilBERT	100K	0.5055	0.5034	0.9630	0.6612
	ALBERT	100K	0.5142	0.5087	0.8911	0.6477
	CF	100K	0.5011	0.5011	1.0000	0.6676
	CS	100K	0.5000	1.0000	0.0022	0.0043
	Context	100K	0.4869	0.4934	0.9020	0.6379
	CF + CS	100K	0.5000	0.5022	0.2440	0.3284
	CF + Context	100K	0.4814	0.4896	0.8235	0.6141
	CF + CS + Context	100K	0.4760	0.4863	0.8105	0.6078
CauseNet	CF	197K	0.4989	0.5000	0.0022	0.0043
	CS	197K	0.5000	1.0000	0.0022	0.0043
	Context	197K	0.5284	0.5154	0.9826	**0.6762**
	CF + CS	197K	0.5000	0.6000	0.0065	0.0129
	CF + Context	197K	**0.5786**	0.5506	0.8649	**0.6729**
	CF + CS + Context	197K	**0.5764**	0.5579	0.7451	0.6381

network, e.g., CauseNet. The model automatically sampled negative examples from CauseNet when generating concept embeddings. However, training the model with both true positive and negative samples may generate better concept embeddings, which should be explored in the future.

5.4 Summary

In this chapter, we proposed a novel deep learning framework to answer binary causal questions. We proposed to use the role-oriented causal embeddings of concepts and a semantic similarity technique to discover causality in text. Our proposed approach addresses the challenge of the lack of large datasets for causality detection and demonstrates the effectiveness of role-oriented causal embeddings of concepts in

improving the accuracy of the answering binary causal questions task while keeping a balanced precision and recall. Overall, our proposed approach could be useful for modeling causality in the context of binary causal questions and the proposed approach can be improved further in the future.

Chapter 6

Conclusions and Future Work

This chapter presents a summary of this thesis and discusses future research directions in the context of causal discovery in text.

6.1 Summary of this Thesis

In this thesis, we have proposed a number of deep learning-based techniques to discover causality in text. This section summarizes the areas of causality detection we have focused on in this thesis and outlines our key findings.

- Firstly, we proposed a deep learning-based technique to detect causally related events in Tweets. Social media short text such as Tweets often lack contextual information which is important for causality detection. Our proposed context word extension technique addresses this problem by extracting contextual information from commonsense background knowledge. We have also proposed a sequence-aware event representation technique to extract and represent candidate causal events in tweets. In our experiments, we find that the combination of our novel feature extraction technique and the deep neural network can improve event causality detection in text.

- Secondly, we have proposed a shared causal attention network (SCAN) model to detect ADR words in Tweets. Social media short text such as Tweets are often short and informally written but it is considered to a source of publicly available health-related data. Our goal was to detect ADRs mentioned in Tweets. Adverse reactions are often described as the direct effects of drugs and we exploit this cause-effect relationship between drugs and ADRs mentioned in Tweets. Our proposed model extracts causal features for each word and shares them with the word and POS features. The model applies a multi-head self-attention mechanism on the extracted features to detect ADR words in tweets.

- Thirdly, we have proposed a deep neural network-based approach to answer BCQs. In this work, we have addressed the challenge of training a deep learning model on a relatively smaller high-quality dataset while keeping the performance comparable or better than the state-of-the-art approaches. We have proposed to train a model on automatically extracted publicly available datasets. Also, we have proposed to apply the alternative random walk technique to convert the candidate causal concepts into role-oriented concept embeddings and then use a semantic similarity technique to extract causal concept features. The proposed model combines both concept features and contextual features to learn causal dependency between two candidate concepts. We have demonstrated that the proposed approach can achieve comparable or better accuracy when compared with the benchmark approaches that use a much larger training dataset.

In short, in this thesis, we addressed the causality detection problem in text using deep learning-based techniques. Our proposed techniques utilize background knowledge to discover causal relationships in texts. If the necessary contextual information to detect causality is not available, our proposed approaches compensate that from publicly available knowledge sources.

6.2 Future Research Directions

In this section, we discuss a few future extensions of this thesis.

- Firstly, the event causality detection model could be configured for a domain-specific field. The proposed model uses domain-independent newspaper articles to build the background knowledge base. However, if the model is intended to use for a specific domain, a more domain-specific knowledge base should be used. Also, the automatically extracted causal pairs from social media are not guaranteed to be facts. Often, users spread fake information on social media intentionally or unintentionally. This challenge could be solved by passing causal pairs to an automatic fact checker [149] to validate the prediction.

- Secondly, the causality-driven ADR detection model proposed in this thesis could be extended to build a drub-drug interactions [150] in ADR detection. The ADRs mentioned in a Tweet could be an effect of a single drug or specific combinations of multiple drugs. However, if a Tweet contains only one drug name it is challenging to find if the patient takes any more drugs that could contribute to the ADR. Also, temporal links also need to be considered to determine whether the drugs are taken together or not. This requires a holistic analysis of a patient profile. A future researcher could consider the previous Tweets posted by the same patient and associated metadata to detect drug-drug interactions.

- Thirdly, the framework proposed for answering binary causal questions in this thesis achieves comparable or better accuracy than the existing approaches, but there is some scope for future research. In our experiments, the proposed approach achieves too high precision and too low recall scores in certain cases. We partly addressed this problem by combining the causal and contextual features to answer binary casual questions. Using a training dataset with high-quality cause-effect concept pairs may further improve the performance of answering binary causal questions. In the future, automatic extraction of high-quality causal concept pairs could be explored to avoid manual efforts to prepare such datasets.

Also, the semantic similarity technique proposed in this thesis to map training concepts to input concepts could be improved further by exploring a better language modeling technique with more accurate concept mapping. Finally, the quality of role-oriented causal concept embeddings generated by our proposed approach could be improved by exploring an automatic technique to prepare negative concept pairs to train the embedding generation model.

6.3 Other Relevant Areas

Natural language is dynamic and constantly evolving. NLP is a broach research area and still, there are many unsolved challenges. Bellow, we outline a few relevant research areas that are relevant but out of the scope of this thesis.

- **Chain of causal events** - The cause-effect relationships are considered to be directional and the direction is from causal to effect. However, an effect could act as a cause of another event and in such scenarios, they may form a chain of casual events. An application of causal chains includes the investigation of an incident by a human resource manager. If a chain of events is available, a manager can make informed decisions.

- **Multi-label text categorization** - The aim of multi-label text categorization is to detect label concepts mentioned in texts. There could be more than one concept in a text and the task is to correctly label each concept. For example, in the following customer review comment, "The staffs in the cafe were great but the food was too expensive", the customer mentioned two concepts. The first concept is about hospitality and the other one is about food price. The recent approaches to detect multi-label text categorization apply various deep learning-based techniques such as capsule networks [151], encoder-decoder model [152], and graph neural networks [153].

- **Aspect-based sentiment analysis** - Sentiment detection is a common NLP

task and overall sentiment is detected for a text. However, a text may have different sentiments associated with different aspects or topics. For example, in the review comment, "The staffs in the cafe were great but the food was too expensive", positive sentiment is expressed for the hospitality aspect but the sentiment towards the food price aspect is negative. Recent approaches [154, 155] found transfer learning-based techniques to be effective for aspect-based sentiment detection. The earlier approaches [156, 157] also use hybrid models to capture sentiments associated with aspects. The hybrid models combine the lexicon features with neural network models.

- **Word dependency learning** - Word dependencies denote the grammatical relations of words. Learning word dependency serves as a key part of many NLP tasks. It also helps computers to understand human language. The recent approaches apply belief network and deep neural network [158]. Deep learning-based models are also effective for intra-sentence dependencies [159].

References

[1] S. Khan and S. Parkinson, "Causal Connections Mining Within Security Event Logs," *K-CAP*, p. 38, 2017.

[2] H. Kayesh, M. S. Islam, and J. Wang, "Event causality detection in tweets by context word extension and neural networks," *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pp. 352–357, 2019.

[3] H. Kayesh, M. S. Islam, and J. Wang, "A causality driven approach to adverse drug reactions detection in tweets," in *ADMA*, vol. 11888, pp. 316–330, 2019.

[4] A. Cocos, A. G. Fiks, and A. J. Masino, "Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts," *JAMIA*, vol. 24, no. 4, pp. 813–821, 2017.

[5] J. Bulao, "How Much Data Is Created Every Day in 2021?." https://techjury.net/blog/how-much-data-is-created-every-day/, 2021. [Online; accessed 30-December-2021].

[6] S. Mani and G. F. Cooper, "Causal discovery using A bayesian local causal discovery algorithm," in *MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics, San Francisco, California, USA*, 2004.

[7] Z. Luo, Y. Sha, K. Q. Zhu, S.-w. Hwang, and Z. Wang, "Commonsense causal reasoning between short texts," *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 421–431, 2016.

[8] S. Sasaki, S. Takase, N. Inoue, N. Okazaki, and K. Inui, "Handling Multiword Expressions in Causality Estimation," *IWCS*, 2017.

[9] S. Doan, E. W. Yang, S. S. Tilak, P. W. Li, D. S. Zisook, and M. Torii, "Extracting health-related causality from twitter messages using natural language processing," *BMC Med. Inf. & Decision Making*, vol. 19-S, no. 3, pp. 71–77, 2019.

[10] 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.

[11] K. Radinsky, S. Davidovich, and S. Markovitch, "Learning causality for news events prediction," *WWW Conference*, pp. 909–918, 2012.

[12] M. Roemmele and A. Gordon, "An encoder-decoder approach to predicting causal relations in stories," in *Proceedings of the First Workshop on Storytelling*, pp. 50–59, 2018.

[13] R. Girju, "Automatic detection of causal relations for question answering," in *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pp. 76–83, 2003.

[14] Z. Li, X. Ding, T. Liu, J. E. Hu, and B. V. Durme, "Guided generation of cause and effect," *CoRR*, vol. abs/2107.09846, 2021.

[15] D. Marcu and A. Echihabi, "An unsupervised approach to recognizing discourse relations," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pp. 368–375, ACL, 2002.

[16] E. Charniak, "A maximum-entropy-inspired parser," in *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pp. 132–139, ACL, 2000.

[17] D. Chang and K. Choi, "Causal relation extraction using cue phrase and lexical pair probabilities," in *Natural Language Processing - IJCNLP 2004, First International*

Joint Conference, Hainan Island, China, March 22-24, 2004, Revised Selected Papers, vol. 3248 of *Lecture Notes in Computer Science*, pp. 61–70, Springer, 2004.

[18] D. I. Moldovan, S. M. Harabagiu, R. Girju, P. Morarescu, V. F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan, "LCC tools for question answering," in *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002* (E. M. Voorhees and L. P. Buckland, eds.), vol. 500-251 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 2002.

[19] E. Blanco, N. Castell, and D. Moldovan, "Causal Relation Extraction," *International Conference on Language Resources and Evaluation (LREC)*, pp. 310–313, 2008.

[20] I. H. Witten and E. Frank, *Data mining - practical machine learning tools and techniques, Second Edition.* The Morgan Kaufmann series in data management systems, Morgan Kaufmann, 2005.

[21] H. Bunke and A. Sanfeliu, *Syntactic and Structural Pattern Recognition-Theory and Applications*, vol. 7. World Scientific, 1990.

[22] 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.

[23] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*, pp. 231–243, Springer, 2010.

[24] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining - A general survey and comparison," *SIGKDD Explor.*, vol. 2, no. 1, pp. 58–64, 2000.

[25] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *ICML*, pp. 282–289, 2001.

[26] Q. X. Do, Y. S. Chan, and D. Roth, "Minimally Supervised Event Causality Identification," *EMNLP*, pp. 294–303, 2011.

[27] D. Roth and W.-t. Yih, "Global inference for entity and relation identification via a linear programming formulation," *Introduction to statistical relational learning*, pp. 553–580, 2007.

[28] R. Prasad, E. Miltsakaki, N. Dinesh, A. Lee, A. Joshi, R. . Prasad, E. . Miltsakaki, N. . Dinesh, A. . Lee, A. . Joshi, L. . Robaldo, and B. L. Webber, "The Penn Discourse Treebank 2.0 Annotation Manual," *IRCS Technical Reports Series*, p. 203, 2007.

[29] M. Roemmele, C. A. Bejan, and A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning," in *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*, AAAI, 2011.

[30] A. Gordon and R. Swanson, "Identifying personal stories in millions of weblog entries," in *Third international conference on weblogs and social media, data challenge workshop, San Jose, CA*, vol. 46, pp. 16–23, 2009.

[31] B. Rink and S. M. Harabagiu, "UTD: classifying semantic relations by combining lexical and semantic resources," in *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010* (K. Erk and C. Strapparava, eds.), pp. 256–259, The Association for Computer Linguistics, 2010.

[32] T. Brants, "Web 1t 5-gram version 1," *http://www. ldc. upenn. edu/Catalog/CatalogEntry. jsp? catalogId= LDC2006T13*, 2006.

[33] S. Pal, P. Pakray, D. Das, and S. Bandyopadhyay, "JU: A supervised approach to identify semantic relations from paired nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala*

University, Uppsala, Sweden, July 15-16, 2010 (K. Erk and C. Strapparava, eds.), pp. 206–209, The Association for Computer Linguistics, 2010.

[34] S. Zhao, T. Liu, S. Zhao, Y. Chen, and J. Nie, "Event causality extraction based on connectives analysis," *Neurocomputing*, vol. 173, pp. 1943–1950, 2016.

[35] M. Riaz and R. Girju, "Recognizing Causality in Verb-Noun Pairs via Noun and Verb Semantics," *EACL-CAtoCL*, pp. 48–57, 2014.

[36] X. Yang and K. Mao, "Multi level causal relation identification using extended features," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7171–7181, 2014.

[37] C. Hidey and K. McKeown, "Identifying causal relations using parallel wikipedia articles," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, The Association for Computer Linguistics, 2016.

[38] S. Bethard and J. H. Martin, "Learning semantic links from a corpus of parallel temporal and causal relations," in *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers*, pp. 177–180, The Association for Computer Linguistics, 2008.

[39] B. Rink, C. A. Bejan, and S. M. Harabagiu, "Learning textual graph patterns to detect causal event relations," in *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, May 19-21, 2010, Daytona Beach, Florida, USA* (H. W. Guesgen and R. C. Murray, eds.), AAAI Press, 2010.

[40] P. Mirza, "Extracting Temporal and Causal Relations between Events," *ACL Student Research Workshop*, pp. 10–17, 2014.

[41] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, *et al.*, "The timebank corpus," in *Corpus linguistics*, vol. 2003, p. 40, 2003.

[42] D. Graff, *The aquaint corpus of English news text:[content copyright] Portions© 1998-2000 New York Times, Inc.,© 1998-2000 Associated Press, Inc.,© 1996-2000 Xinhua News Service*. Linguistic Data Consortium, 2002.

[43] P. Mirza and S. Tonelli, "CATENA: causal and temporal relation extraction from natural language texts," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan* (N. Calzolari, Y. Matsumoto, and R. Prasad, eds.), pp. 64–75, ACL, 2016.

[44] Q. Ning, Z. Feng, H. Wu, and D. Roth, "Joint reasoning for temporal and causal relations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (I. Gurevych and Y. Miyao, eds.), pp. 2278–2288, Association for Computational Linguistics, 2018.

[45] Q. Ning, Z. Feng, H. Wu, and D. Roth, "Joint reasoning for temporal and causal relations," *CoRR*, vol. abs/1906.04941, 2019.

[46] M. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," *Mach. Learn.*, vol. 88, no. 3, pp. 399–431, 2012.

[47] T. Cassidy, B. McDowell, N. Chambers, and S. Bethard, "An annotation framework for dense event ordering," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pp. 501–506, The Association for Computer Linguistics, 2014.

[48] P. Mirza and S. Tonelli, "An analysis of causality between events and its relation to temporal information," in *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland* (J. Hajic and J. Tsujii, eds.), pp. 2097–2106, ACL, 2014.

[49] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," *CoRR*, vol. abs/1911.10422, 2019.

[50] J. Pustejovsky, J. M. Castaño, R. Ingria, R. Saurí, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, "Timeml: Robust specification of event and temporal expressions in text," in *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA* (M. T. Maybury, ed.), pp. 28–34, AAAI Press, 2003.

[51] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010* (T. Kobayashi, K. Hirose, and S. Nakamura, eds.), pp. 1045–1048, ISCA, 2010.

[52] B. Rosario and M. A. Hearst, "Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2001, Pittsburgh, PA USA, June 3-4, 2001*, ACL, 2001.

[53] P. Leixo, T. A. S. Pardo, *et al.*, "Cstnews: um corpus de textos jornalísticos anotados segundo a teoria discursiva multidocumento cst (cross-document structure theory)," 2008.

[54] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. F. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016* (K. Knight, A. Nenkova, and O. Rambow, eds.), pp. 839–849, The Association for Computational Linguistics, 2016.

[55] T. Dasgupta, R. Saha, L. Dey, and A. Naskar, "Automatic extraction of causal relations from text using linguistically informed deep neural networks," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018* (K. Komatani, D. J. Litman, K. Yu, L. Cavedon, M. Nakano, and A. Papangelis, eds.), pp. 306–316, Association for Computational Linguistics, 2018.

[56] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[57] M. Nauta, D. Bucur, and C. Seifert, "Causal discovery with attention-based convolutional neural networks," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 312–340, 2019.

[58] P. Li and K. Mao, "Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts," *Expert Syst. Appl.*, vol. 115, pp. 512–523, 2019.

[59] D. Bollegala, S. Maskell, R. Sloane, J. Hajne, and M. Pirmohamed, "Causality patterns for detecting adverse drug reactions from social media: Text mining approach," *JMIR Public Health and Surveillance*, vol. 4, no. 2, 2018.

[60] C. Kruengkrai, 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*, pp. 3466–3473, 2017.

[61] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3642–3649, 2012.

[62] J. Oh, K. Torisawa, C. Kruengkrai, R. Iida, and J. Kloetzer, "Multi-column convolutional neural networks with causality-attention for why-question answering," in *Proceedings of the Tenth ACM International Conference on Web*

Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017 (M. de Rijke, M. Shokouhi, A. Tomkins, and M. Zhang, eds.), pp. 415–424, ACM, 2017.

[63] K. Kadowaki, R. Iida, K. Torisawa, J. Oh, and J. Kloetzer, "Event causality recognition exploiting multiple annotators' judgments and background knowledge," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), pp. 5815–5821, Association for Computational Linguistics, 2019.

[64] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4171–4186, Association for Computational Linguistics, 2019.

[65] V. Khetan, R. R. Ramnani, M. Anand, S. Sengupta, and A. E. Fano, "Causal-bert : Language models for causality detection between events expressed in text," *CoRR*, vol. abs/2012.05453, 2020.

[66] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *J. Biomed. Informatics*, vol. 45, no. 5, pp. 885–892, 2012.

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[68] C. Buck, K. Heafield, and B. van Ooyen, "N-gram counts and language models from the common crawl," in *Proceedings of the Ninth International Conference on*

Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014 (N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, eds.), pp. 3579–3584, European Language Resources Association (ELRA), 2014.

[69] B. Yu, Y. Li, and J. Wang, "Detecting causal language use in science findings," *EMNLP-IJCNLP*, pp. 4663–4673, 2019.

[70] E. Rahimtoroghi, E. Hernandez, and M. A. Walker, "Learning Fine-Grained Knowledge about Contingent Relations between Everyday Events," *SIGDIAL*, pp. 350–359, 2016.

[71] K. Roberts and S. M. Harabagiu, "Detecting new and emerging events in streaming news documents," *Int. J. Semantic Computing*, vol. 5, no. 4, pp. 407–431, 2011.

[72] P. Mirza, "Extracting temporal and causal relations between events," *CoRR*, vol. abs/1604.08120, 2016.

[73] A. Balashankar, S. Chakraborty, S. Fraiberger, and L. Subramanian, "Identifying predictive causal factors from news streams," in *EMNLP-IJCNLP* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), pp. 2338–2348, Association for Computational Linguistics, 2019.

[74] H. Kayesh, M. S. Islam, and J. Wang, "A causality driven approach to adverse drug reactions detection in tweets," *International Conference on Advanced Data Mining and Applications (ADMA)*, pp. 316–330, 2019.

[75] W. Ali, W. Zuo, R. Ali, X. Zuo, and G. Rahman, "Causality mining in natural languages using machine and deep learning techniques: A survey," *Applied Sciences*, vol. 11, no. 21, 2021.

[76] C. C. Yang, H. Yang, L. Jiang, and M. Zhang, "Social media mining for drug safety signal detection," in *International Workshop on Smart Health and Wellbeing*, pp. 33–40, 2012.

[77] Y. Ji, H. Ying, P. Dews, A. Mansour, J. Tran, R. E. Miller, and R. M. Massanari, "A potential causal association mining algorithm for screening adverse drug reactions in postmarketing surveillance," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 428–437, 2011.

[78] S. Evans, P. C. Waller, and S. Davis, "Use of proportional reporting ratios (prrs) for signal generation from spontaneous adverse drug reaction reports," *Pharmacoepidemiology and drug safety*, vol. 10, no. 6, pp. 483–486, 2001.

[79] X. Qin, T. Kakar, S. Wunnava, E. A. Rundensteiner, and L. Cao, "Maras: signaling multi-drug adverse reactions," in *KDD*, pp. 1615–1623, 2017.

[80] H. Yang and C. C. Yang, "Using health-consumer-contributed data to detect adverse drug reactions by association mining with temporal analysis," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 4, 2015.

[81] W.-Y. Lin, H.-Y. Li, J.-W. Du, W.-Y. Feng, and C.-F. Lo, "Iadrs: Towards a web-based interactive adverse drug reaction analyzing system," *SIGHIT Rec.*, vol. 2, no. 1, p. 22, 2012.

[82] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[83] T. Huynh, Y. He, A. Willis, and S. Rüger, "Adverse drug reaction classification with deep neural networks," in *COLING*, pp. 877–887, 2016.

[84] C. Wu, F. Wu, Z. Yuan, J. Liu, Y. Huang, and X. Xie, "Msa: Jointly detecting drug name and adverse drug reaction mentioning tweets with multi-head self-attention," in *WSDM*, pp. 33–41, 2019.

[85] J. Chu, W. Dong, K. He, H. Duan, and Z. Huang, "Using neural attention networks to detect adverse medical events from electronic health records," *J. Biomed. Informatics*, vol. 87, pp. 118–130, 2018.

[86] T. Zhang, H. Lin, Y. Ren, L. Yang, B. Xu, Z. Yang, J. Wang, and Y. Zhang, "Adverse drug reaction detection via a multihop self-attention mechanism," *BMC Bioinform.*, vol. 20, no. 1, pp. 479:1–479:11, 2019.

[87] T. Zhang, H. Lin, B. Xu, Y. Ren, Z. Yang, J. Wang, and X. Duan, "Gated iterative capsule network for adverse drug reaction detection from social media," in *International Conference on Bioinformatics and Biomedicine, BIBM*, pp. 387–390, IEEE, 2020.

[88] Y. Luo, "Recurrent neural networks for classifying relations in clinical notes," *J. Biomed. Informatics*, vol. 72, pp. 85–95, 2017.

[89] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[90] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[91] Q. Song, B. Li, and Y. Xu, "Research on adverse drug reaction recognitions based on conditional random field," in *International Conference on Business and Information Management*, pp. 97–101, 2017.

[92] K. O'Connor, P. Pimpalkhute, A. Nikfarjam, R. Ginn, K. L. Smith, and G. Gonzalez, "Pharmacovigilance on twitter? mining tweets for adverse drug reactions," in *AMIA annual symposium proceedings*, vol. 2014, p. 924, 2014.

[93] A. Nikfarjam, A. Sarker, K. O'connor, R. Ginn, and G. Gonzalez, "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features," *JAMIA*, vol. 22, no. 3, pp. 671–681, 2015.

[94] A. Metke-Jimenez and S. Karimi, "Concept identification and normalisation for adverse drug event discovery in medical forums," in *Proceedings of the First International Workshop on Biomedical Data Integration and Discovery (BMDID 2016) co-located with The 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 17, 2016* (C. Tao, G. Jiang, D. Song, J. Heflin, and F. Schilder, eds.), vol. 1709 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016.

[95] M. Z. Sh zulfatmi, T. E. V elvtutubalina, T. A. E alex, M. Z. Sh, T. E. V, and T. A. E, "IdentIfyIng dIsease-related expressIons In revIews UsIng CondItIonal random fIelds," in *Proceedings of the International Conference*, 2017.

[96] S. Chowdhury, C. Zhang, and P. S. Yu, "Multi-task pharmacovigilance mining from social media posts," *WWW*, 2018.

[97] E. Flórez, F. Precioso, M. Riveill, and R. Pighetti, "Named entity recognition using neural networks for clinical notes," in *Proceedings of the 1st International Workshop on Medication and Adverse Drug Event Detection*, vol. 90 of *Proceedings of Machine Learning Research*, pp. 7–15, PMLR, 2018.

[98] E. Flórez, F. Precioso, R. Pighetti, and M. Riveill, "Deep learning for identification of adverse drug reaction relations," in *Proceedings of the 2019 International Symposium on Signal Processing Systems*, pp. 149–153, ACM, 2019.

[99] E. Tutubalina and S. Nikolenko, "Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews," *Journal of healthcare engineering*, vol. 2017, 2017.

[100] P. Ding, X. Zhou, X. Zhang, J. Wang, and Z. Lei, "An attentive neural sequence labeling model for adverse drug reactions mentions extraction," *IEEE Access*, vol. 6, pp. 73305–73315, 2018.

[101] E. El-allaly, M. Sarrouti, N. Ennahnahi, and S. O. E. Alaoui, "An adverse drug effect mentions extraction method based on weighted online recurrent extreme learning machine," *Comput. Methods Programs Biomed.*, vol. 176, pp. 33–41, 2019.

[102] D. Mahata, S. Anand, H. Zhang, S. Shahid, L. Mehnaz, Y. Kumar, and R. Shah, "Midas@ smm4h-2019: identifying adverse drug reactions and personal health experience mentions from twitter," in *Proceedings of the fourth social media mining for health applications (# SMM4H) workshop & shared task*, pp. 127–132, 2019.

[103] Z. Miftahutdinov, I. Alimova, and E. Tutubalina, "Kfu nlp team at smm4h 2019 tasks: Want to extract adverse drugs reactions from tweets? bert to the rescue," in

Proceedings of the fourth social media mining for health applications (# SMM4H) workshop & shared task, pp. 52–57, 2019.

[104] X. Zhao, Y. Xiong, and B. Tang, "Hitsz-icrc: A report for smm4h shared task 2020-automatic classification of medications and adverse effect in tweets," in *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pp. 146–149, 2020.

[105] Z. Li, Z. Yang, L. Wang, Y. Zhang, H. Lin, and J. Wang, "Lexicon knowledge boosted interaction graph network for adverse drug reaction recognition from social media," *IEEE J. Biomed. Health Informatics*, vol. 25, no. 7, pp. 2777–2786, 2021.

[106] W. Zhang, Z. Kuang, P. Peissig, and D. Page, "Adverse drug reaction discovery from electronic health records with deep neural networks," in *The ACM Conference on Health, Inference, and Learning*, p. 30–39, 2020.

[107] C. Pandey, Z. Ibrahim, H. Wu, E. Iqbal, and R. Dobson, "Improving rnn with attention and embedding for adverse drug reactions," in *International Conference on Digital Health*, p. 67–71, 2017.

[108] A. N. Jagannatha and H. Yu, "Bidirectional RNN for medical event detection in electronic health records," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016* (K. Knight, A. Nenkova, and O. Rambow, eds.), pp. 473–482, The Association for Computational Linguistics, 2016.

[109] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (A. Moschitti, B. Pang, and W. Daelemans, eds.), pp. 1724–1734, ACL, 2014.

[110] S. Wunnava, X. Qin, T. Kakar, E. A. Rundensteiner, and X. Kong, "Bidirectional LSTM-CRF for adverse drug event tagging in electronic health records," in *Proceedings of the 1st International Workshop on Medication and Adverse Drug Event Detection, 4 May 2018*, vol. 90 of *Proceedings of Machine Learning Research*, pp. 48–56, PMLR, 2018.

[111] O. Hassanzadeh, D. Bhattacharjya, M. Feblowitz, K. Srinivas, M. Perrone, S. Sohrabi, and M. Katz, "Answering binary causal questions through large-scale text mining: An evaluation using cause effect pairs from human experts," *IJCAI*, pp. 5003–5009, 2019.

[112] H. Kayesh, M. S. Islam, J. Wang, S. Anirban, A. Kayes, and P. Watters, "Answering binary causal questions: A transfer learning based approach," *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[113] D. Preoţiuc-Pietro, Y. Liu, D. Hopkins, and L. Ungar, "Beyond binary labels: Political ideology prediction of twitter users," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 729–740, 2017.

[114] M. Hasanuzzaman, S. Kamila, M. Kaur, S. Saha, and A. Ekbal, "Temporal orientation of tweets for predicting income of users," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 659–665, 2017.

[115] Y. Liu, L. Zhang, L. Nie, Y. Yan, and D. S. Rosenblum, "Fortune teller: Predicting your career path," *AAAI*, p. 201–207, 2016.

[116] A. Ritter, S. Clark, and O. Etzioni, "Named Entity Recognition in Tweets: An Experimental Study," *EMNLP*, pp. 1524–1534, 2011.

[117] H. Kayesh, M. S. Islam, and J. Wang, "On event causality detection in tweets," *arXiv preprint arXiv:1901.03526*, 2019.

[118] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," *EMNLP*, pp. 740–750, 2014.

[119] B. Jang and J. Yoon, "Characteristics analysis of data from news and social network services," *IEEE Access*, vol. 6, pp. 18061–18073, 2018.

[120] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?," *NewsIR*, pp. 42–47, 2016.

[121] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *NIPS*, pp. 3111–3119, 2013.

[122] J. Sultana, P. Cutroneo, and G. Trifirò, "Clinical and economic burden of adverse drug reactions," *Journal of pharmacology & pharmacotherapeutics*, vol. 4, no. Suppl1, p. S73, 2013.

[123] S. Karimi, C. Wang, A. Metke-Jimenez, R. Gaire, and C. Paris, "Text and data mining techniques in adverse drug reaction detection," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–39, 2015.

[124] C. C. Yang, L. Jiang, H. Yang, and X. Tang, "Detecting signals of adverse drug reactions from health consumer contributed content in social media," in *Proceedings of ACM SIGKDD Workshop on Health Informatics*, ACM, 2012.

[125] F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations," in *Proceedings of the Workshop on Noisy User-generated Text*, pp. 146–153, 2015.

[126] A. Cocos, A. G. Fiks, and A. J. Masino, "Reply to comment on:"deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts"," *Journal of the American Medical Informatics Association*, vol. 26, no. 6, pp. 580–581, 2019.

[127] A. Magge, A. Sarker, A. Nikfarjam, and G. Gonzalez-Hernandez, "Comment on:"deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts"," *Journal of the American Medical Informatics Association*, vol. 26, no. 6, pp. 577–579, 2019.

[128] J. Dietrich, L. M. Gattepaille, B. A. Grum, L. Jiri, M. Lerch, D. Sartori, and A. Wisniewski, "Adverse events in twitter-development of a benchmark reference dataset: Results from imi web-radr," *Drug safety*, pp. 1–12, 2020.

[129] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Neural networks for machine learning, Coursera lecture 6e*, 2012.

[130] L. V. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. S. Batra, P. V. Kamesam, and R. Kothari, "Information extraction from biomedical literature: methodology, evaluation and an application," in *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pp. 410–417, ACM, 2003.

[131] R. T.-H. Tsai, S.-H. Wu, W.-C. Chou, Y.-C. Lin, D. He, J. Hsiang, T.-Y. Sung, and W.-L. Hsu, "Various criteria in the evaluation of biomedical named entity recognition," *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–8, 2006.

[132] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2018.

[133] S. Heindorf, Y. Scholten, H. Wachsmuth, A. N. Ngomo, and M. Potthast, "Causenet: Towards a causality graph extracted from the web," in *CIKM*, pp. 3023–3030, ACM, 2020.

[134] M. Khosla, J. Leonhardt, W. Nejdl, and A. Anand, "Node representation learning for directed graphs," in *ECML-PKDD*, pp. 395–411, 2019.

[135] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for english," in *EMNLP*, pp. 169–174, 2018.

[136] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[137] H. Kayesh, M. S. Islam, J. Wang, A. S. M. Kayes, and P. A. Watters, "A deep learning model for mining and detecting causally related events in tweets," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 2, 2022.

[138] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?," in *NewsIR*, pp. 42–47, 2016.

[139] S. Sohrabi, A. V. Riabov, M. Katz, and O. Udrea, "An ai planning solution to scenario generation for enterprise risk management," in *AAAI*, 2018.

[140] S. Sohrabi, M. Katz, O. Hassanzadeh, O. Udrea, M. D. Feblowitz, and A. Riabov, "Ibm scenario planning advisor: Plan recognition as ai planning in practice," *AI Communications*, no. Preprint, pp. 1–13, 2019.

[141] NATO, "Stratigic Foresight Analysis 2017 report." https://www.act.nato.int/publications-ffao, 2017. [Online; accessed February 21, 2019].

[142] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *SEW*, pp. 94–99, 2009.

[143] H. Kayesh, M. S. Islam, and J. Wang, "Event causality detection in tweets by context word extension and neural networks," in *PDCAT*, pp. 352–357, 2019.

[144] H. Kayesh, M. Islam, and J. Wang, "On event causality detection in tweets," *arXiv preprint arXiv:1901.03526*, 2019.

[145] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[146] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[147] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[148] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[149] P. Atanasova, P. Nakov, L. Màrquez, A. Barrón-Cedeño, G. Karadzhov, T. Mihaylova, M. Mohtarami, and J. R. Glass, "Automatic fact-checking using context and discourse information," *ACM J. Data Inf. Qual.*, vol. 11, no. 3, pp. 12:1–12:27, 2019.

[150] A. D. Rodrigues, *Drug-drug interactions*. CRC Press, 2019.

[151] B. Chen, X. Huang, L. Xiao, and L. Jing, "Hyperbolic Capsule Networks for Multi-Label Classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3115–3124, Association for Computational Linguistics (ACL), jul 2020.

[152] W. Liao, Y. Wang, Y. Yin, X. Zhang, and P. Ma, "Improved sequence generation model for multi-label classification via CNN and initialized fully connection," *Neurocomputing*, vol. 382, pp. 188–195, mar 2020.

[153] Y. Zhang, Y. Meng, J. Huang, F. F. Xu, X. Wang, and J. Han, "Minimally supervised categorization of text with metadata," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1231–1240, 2020.

[154] X. Fang and J. Tao, "A transfer learning based approach for aspect based sentiment analysis," in *Sixth International Conference on Social Networks Analysis, Management and Security, SNAMS 2019, Granada, Spain, October 22-25, 2019* (M. A. Alsmirat and Y. Jararweh, eds.), pp. 478–483, IEEE, 2019.

[155] D. Meškelė and F. Frasincar, "ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and

a regularized neural attention model," *Information Processing & Management*, vol. 57, no. 3, p. 102211, 2020.

[156] N. Zainuddin, A. Selamat, and R. Ibrahim, "Hybrid sentiment classification on twitter aspect-based sentiment analysis," *Applied Intelligence*, vol. 48, no. 5, pp. 1218–1232, 2018.

[157] D. Meškelė and F. Frasincar, "Aldona: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalised domain ontology and a neural attention model," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 2489–2496, 2019.

[158] I. Chaturvedi, Y. S. Ong, I. W. Tsang, R. E. Welsch, and E. Cambria, "Learning word dependencies in text by means of a deep recurrent belief network," *Knowledge-Based Systems*, vol. 108, pp. 144–154, 2016.

[159] D. Guo, B. Chen, R. Lu, and M. Zhou, "Recurrent Hierarchical Topic-Guided RNN for Language Generation," in *Proceedings of the 37th International Conference on Machine Learning, PMLR*, pp. 3810–3821, PMLR, nov 2020.