A Study of Extracting Causal Relationships from Text

Pranav Gujarathi, Manohar Reddy, Neha Tayade, and Sunandan Chakraborty

Indiana University, IN 47404, USA,
gujarathi.pranav@gmail.com, {peddires,ntayade,sunchak@}iu.edu

Abstract. Discovering causal knowledge is an important aspect of much scientific research and such findings are often recorded in scholarly articles. Automatically identifying such knowledge from article text can be a useful tool and can act as an impetus for further research on those topics. Numerous applications, including building a causal knowledge graph, making pipelines for root cause analysis, discovering opportunities for drug discovery, and overall, a scalable building block towards turning large pieces of text into organized information can be built following such an approach. However, it requires robust methods to identify and aggregate causal knowledge from a large set of articles. The main challenge in designing new methods is the absence of a large labeled dataset. As a result, existing methods trained on existing datasets with limited size and variations in linguistic pattern, are unable to generalize well on unseen text. In this paper, we explore multiple unsupervised approaches, including a reinforcement learning-based model that learns to identify causal sentences from a small set of labeled sentences. We describe and discuss in detail our experiments for each approach to further encourage exploration of methods that can be re-utilized for different tasks as well, as opposed to simply exploring a supervised learning process which although superior in performance lacks the versatility to be re-purposed for slightly different tasks. We evaluate our methods on a custom-created dataset and show unique techniques to extract cause-effect relationships from the English language.

Keywords: causal relations, reinforcement learning, relationship extraction

1 Introduction

Causal relationships depict important knowledge across many different fields, including medicine, health, economics, and public policy. Researchers in these fields design and conduct experiments to verify causality between two events and publish their findings in research articles. Such research articles record the discovery of new causal relationships or some new conditions of existing causal relationships providing new knowledge in the field. Causal relationships expressed in the text provide a unique opportunity to discover new causal knowledge and capture the fundamentally dialogic and dynamic nature of knowledge. New research findings may refute existing causal relationships, or new conditions may redefine such relationships. Thus, mining such relationships from peer-reviewed articles will help to create a knowledge of causal relationships, that may include, conflicting findings, inconsistent discoveries, refutations, contradictions, reinforcements, or confirmations, all changing over time [1,2]. Hence, such a knowledge base can help to help to capture the dynamism exhibited by causal relations. An example

of a causal relationship is expressed in text is shown in Fig. 1. Extracting such relations from text, however, is not trivial, as expressing causality through natural language may take many different forms. For example, for two events *A* and *B* the causal relationships between *A* and *B* can be expressed using active voice (e.g. *"A causes B"*), using passive voice (*"B is caused by A"*) or using synonymous expressions (e.g. *"A leads to B"*)and so on.

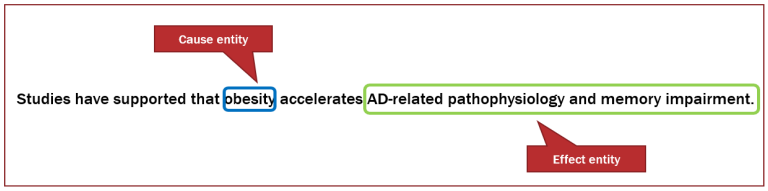

Fig. 1. An example of a causal relationship in text

This paper focuses on the self-learning method of detection and extraction of causal relationships from the open-domain text for analytical and predictive applications. Humans are predisposed to understand counterfactual and situational information by inferring cause-and-effect patterns from statements. In articles, documents, and many other text resources, causal reasonings generally appear in the form of descriptive, inductive, or abductive associations between the agent and the act. These variations in the strength of connections make it challenging for machine-based techniques to extract association effects using any particular supervised or rule-based method. To illustrate this further, we can say that sometimes the information indicators are clear or explicit by the usage of words like- *caused by, led to, influences*, etc. in the text which leads to an easy discovery of the association. But during other times, this relationship has more implicit or ambiguous indications. For instance, words like *show that, trigger, arose*, etc. can have multiple connotations which can only be apprehended using complex and self-learned methods.

Operationally, causal relationships denote how different events and entities should be perceived in relation to each other and can be linguistically modeled by discovering the type of reasoning- common cause-effect, causal chains, or homeostasis, between the statements. By assessing the type of dependence, covariance, or dynamism in the text, we can try to find out how models can learn these modes distinctly and apply them to new samples passed to it.

There are many challenges in extracting causal relationships from the text. A simple method can be to use a pre-trained dependency parser and a rule-based system that captures causal relationships through grammatical constructs. However, multiple experiments later, it is found that generalizing a rule-based formula from the dependencies is not possible and a deeper understanding of the grammatical structure of the language is necessary, as well as understanding the flow of context.

Through the course of our work, we explored multiple approaches, both supervised and unsupervised (refer Fig. 2). While unsupervised methods lack the pinpoint accuracy and ability to be deployed in a practical application without further development, these

approaches provide a template for building NLP solutions without the need for hand-annotated data halting the progress. This is a significant improvement from previous work [12], where all such models were trained on existing causal relationship datasets, such as SemEval-2010 [3] and Adverse Drug Effect (ADE) [4] with several limitations, such as small size and minimal variation in how the relationship is expressed. As a result, these models' performance drops when they are applied to real-world sentences.

On the other hand, unsupervised approaches can be scaled to multiple different applications and are a step in the direction of generalized artificial intelligence. In this paper, we propose four unsupervised approaches that can be applied to larger and more generic text to detect causal relations and overcome the limitations of the existing datasets. In one of our unsupervised approaches, we have tried the Actor-Critic (A2C) Reinforcement Learning (RL) [7] method where a reward function that evaluated the quality of predictions is optimized to improve the predictions from a state of random walk toward convergence. In our exploration of related work, we found that our approach to setting up the RL problem in a Natural Language scenario was unique and can be repurposed to many other various NLP problems. We compare our unsupervised methods with two more supervised methods that are similar in terms of the annotated dataset used, the challenges, and pre-processing of the text, but differ in finer aspects of setting up the training process vis-a-vis the dependent and independent variable. Both the supervised approaches utilize light transformer-based AlBert [10] architecture while adding a few additional layers depending on the dimensions of input and output variables.

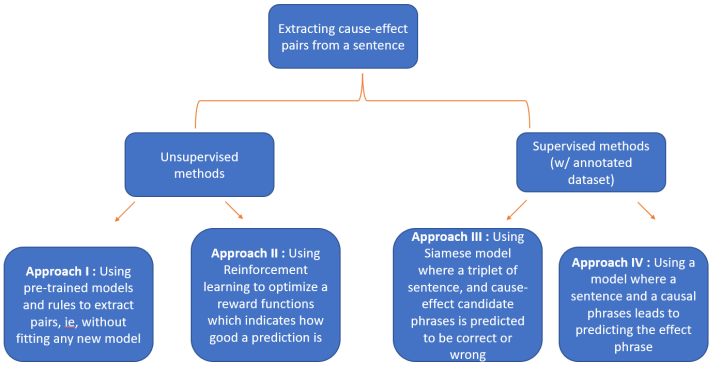

Fig. 2. A hierarchical representation of different approaches explored to extract causal relationships from text

2 Related Work

Several combinations of linguistic, rule-based, machine and deep learning techniques have been used in the past to detect causal semantics from text. These past works can

be considered as the criterion to further explore and modify potentially well performing areas to achieve better results. Bayesian methods [14] used to extract and filter cause-effect pairs by accounting for lexical and semantic features that analyze most frequent relations in dependency trees of sentences resulting into precision as high as 71% for SemEval dataset. Another similar Information theory Bayesian approach [15] calculates probabilities on likelihood between drugs and their harm and then learns parameters using Artificial Neural Networks. Egami et al (2018) presents a generalizable Codebook function(g) [16] fitting onto several models, to establish a link between higher dimensional text to lower dimensional representation and group them for making inferences and estimations of treatment effects. A scenario based Supervised method [17] probabilistically extracts binary semantic relation features to constitute for event causality scores ranked by SVM.

Lexical models with task-specific causal embeddings [19] relate answers to questions by comparing noun phrases to causal mentions using Causal Convolutional Neural Networks is another way to use causal inferencing in QA applications. Software prototypes [20] to extract cause, effects, interaction signs built with NLP on annotated corpuses of research papers using hypothesis parsing and a set of rules have resulted in a significant F1 score between 0.71 to 0.90. As an extension of this method, extraction of a list of multiword expressions [21] is possible based on lexico-syntactic patterns and coreference relations, and estimate causal relations using statistical state-of-the-art Pointwise Mutual Information metric on Choice of Possible Alternatives (COPA) with an accuracy of 72%. The CausalTriads model is a more comprehensive approach on capturing transitive dependencies [22] to discover unseen causal relations and generate new causal hypothesis. It uses four structures to represent the rules of causal transitivity laid out by a factor graph model. An interesting and novel method to address explanations of a causal event is by using time series [23] to search relations between cause-effect and construct a chain between them to generate an explanation. The prior work of feature extractions, such as n-grams, sentiments, topic, etc., is translated into a CGraph of sequential causal entities requiring commonsense causative knowledge base with efficient reasoning. Also, by describing a causal word to be of simple, resultative, or instrumental can be helpful in structuring the relational rules and semantic constraints on parts-of-speech mappings in sentences as discussed in the paper for text mining [24]. An approach similar to the one presented in this paper, includes lexical pair probabilities [25] and cue phrases learned from raw corpus in an unsupervised manner to extract events followed by their relation extraction using simple Naïve Bayes classifiers and Expectation-Maximization optimization. The Causal and Temporal Relations Scheme (CaTeRS) [26] introduces a narrative structure to stories by specializing annotating large sentences in a couple of words to benefit from the rich inferential capabilities that structured knowledge about events can provide using temporal tags.

In recent times, we see the use of deep learning in addressing this problems. Li et al (2021) [18] utilizes cartesian products of entity-mention tags and relation-type tags produced with tag2triplet algorithm that detect causal triplets of two event entities and the relation between them. This is implemented using BiLSTM-CRF and multiheaded self-attention mechanism to capture long-range dependencies outperforming other methods with F-score of 83-85% on different datasets. Causality extraction based on extraction

of lexical semantics and document-clause frequencies [27] from specific literatures make it easier to adopt schemes that determine direction and strengths for the relational frequencies in causal disease network. Another approach [28] uses Deep RL with a CNN and Tree-LSTM networks that model relations during the initial and transition states followed by a penalizing step which increases the penalties for decision-making errors to reduce the problems of unbalanced corpus. The Q-learning algorithm then computes the control policy over that action resulting in a final F1score of up to 78% for the ACE news dataset. Another paper [29] mentions on similar lines, an inference based inductive or deductive causal reasoning algorithm with causal rules which transform start states to next states using and controlling factors generating non-optimal solutions optimized by AI reinforcement learning by varying the learning rate and discount factor at the expense of reduced speed. Alternative to the traditional approaches, the Graph Convolutional Networks for Semantic Role Labeling [30] focuses on annotating sentences using semantic and syntactic dependencies by parsing its structure to model any type of relationship.

Comparison between ML and non-ML paradigms contrast the annotation and hand-coding of linguistic texts to extract features to automated ML techniques involving WordNets, FrameNets for broader coverage areas are discussed in this survey [31]. Techniques for filtering confounding causal estimates [32] simplifies the confounding bias using Doubly-robust algorithms, causal-driven representation learning, regression adjustments, etc. Reinforcement Learning uses external evidence [33] to improve extraction accuracy in domains by using Markov decision process which includes a space of possible states, actions, a reward, and a transition function to dynamically extract causal pairs. Multivariate time series analysis (non-parametric) [34] for graphical modelling efficiently handle non-linear directed acyclic graphs by replacing conditional orthogonality by conditional independence leading to strong Ganger causality. Extraction of explicit and implicit causal relations [35] is a problem for sparse texts, an approach to alleviate this is through explicit causal patterns for explicit cases and associating causal events with causal-agents to generate an evaluative causal valence.

3 Unsupervised Approaches

3.1 Approach I : Using Pre-trained models for inference

Before experimenting with manually annotated data, we explored the possibility of labeling the data using pre-trained models. The HuggingFace library in recent times has not only provided easy to implement transformer models, but also a vast library of pre-trained models, ready to be used as Natural Language Inference tools. For instance, one of the very successfully trained models is the DistillBert [5] Question Answer model trained on the Squad [6] dataset. The unlabeled data is scraped from Pubmed articles based on the occurrence of a list of 'cause phrases' (caused by, lead to, affects, resulting in, etc). It is important to note here, that while one of these words exists in these sentences, it may not imply a causal relationship.

It is also important to define **noun phrases** in the context of our paper. Noun phrases are a continuous sequence of words within a sentence that are compound noun-like

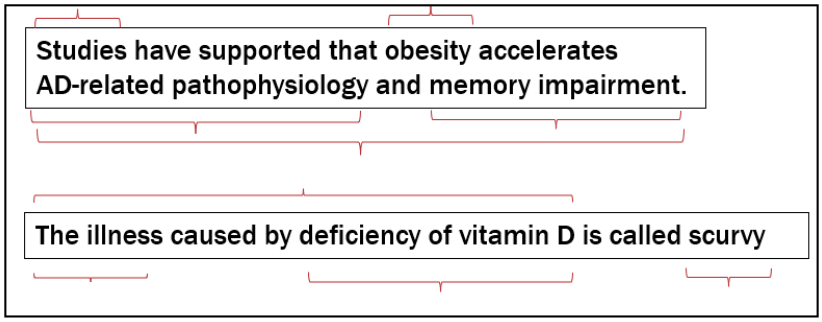

Fig. 3. Examples of Noun Phrases within two sentences. As we can see, there can exist various Noun Phrases within a sentence, ranging from a single noun to a collection of named entities joined by prepositions. We also see that there can be an overlap between Noun Phrases, and one noun phrase can even be a collection of two or more.

entities or complex collections of words, which can be regarded as standalone objects. For instance, consider the sentence - *"The illness caused by deficiency of vitamin D is called scurvy"*. Here while named entities like [deficiency, vitamin D, scurvy, etc] are regarded as noun phrases, the complex phrase *The illness caused by deficiency of vitamin D* is also a noun phrase, since it refers to a single object/entity and can be replaced by the noun *scurvy* in a sentence. Fig. 3 presents a sentence with all possible overlapping noun phrases and Fig. 4 shows how we use all the combinations to build candidate noun phrases to identify cause/effect phrases.

We assume that if one of the phrases, cause or effect is known, we can frame questions to infer the other. Thus, for every cause phrase, we define question templates. For instance, for the cause word 'causes' the set of possible questions will be [What causes {1}?, What does {0} cause?, Does {0} cause {1}?]. We know that both cause and effect belong to the set of noun phrases and that if they occupy the places by {0} and {1} respectively, the questions can be answered with high confidence. For example, the sentence - "*the subjects were exposed to UV irradiation causing a local tissue inflammation*". We extract all noun phrases, and permute all combinations - some questions have only one input word, some have two, we make sure all cases are covered and as many combinations as possible are made with no regard to whether or not the sentence makes any logical sense). Let us say we make ω template questions. The pre-trained QA model takes in two inputs - question and context and returns two inputs - answer phrase and confidence score. Hence for every sentence, we feed in ω different sets of inputs (same context, different question), and finally pick the one with the best confidence score. In this case, the constructed sentence *What is causing local tissue inflammation?* gave the best confidence score with the answer *uv radiation*, and hence we assign *uv radiation* and *local tissue inflammation* respectively as cause and effect. Fig 5 illustrates this process.

We found a few limitations in this approach. First, the noun phrase extraction may not be perfect, and may often just give named entities as opposed to correct noun phrases. Thus the error further carries forward to the next step, affecting overall performance.

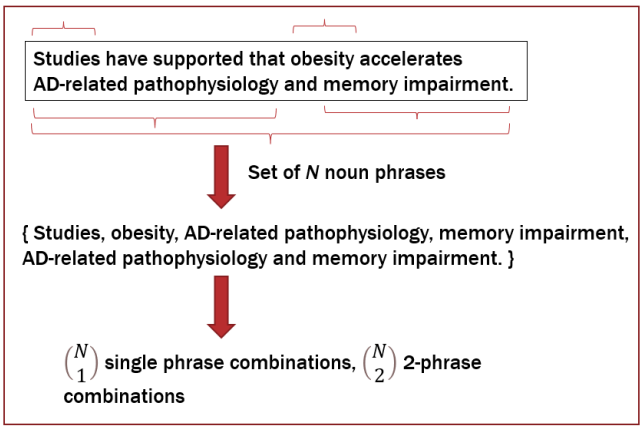

Fig. 4. For Approach I, we first extract all the N noun phrases from the sentence. From this set of N noun phrases, we cycle through $\binom{N}{1}$ combinations of taking one phrase at a time as well as all the $\binom{N}{2}$ combinations taking 2 phrases at a time, generate a sentence from the phrase/phrases and infer if there exists a cause-effect relationship

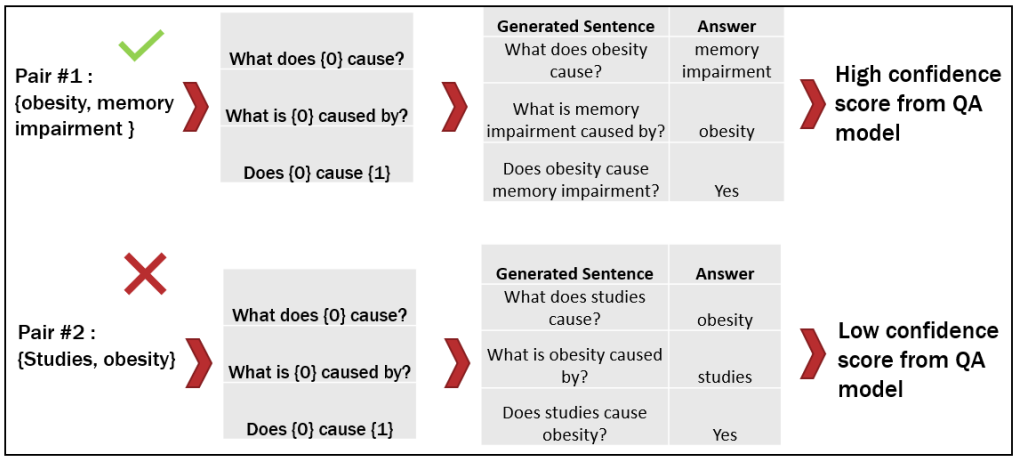

Fig. 5. Illustration of how generated sentences and scores from pre-trained models can distinguished between correctly and incorrectly extracted cause effect phrases, and since the search space is limited (since there are limited noun phrases), this comparative approach can be used for cause effect extraction.

The overall performance is also dependent upon the performance of the QA model. A QA model can answer questions correctly only when the context is provided, however, the way some noun phrases are extracted this may not be true. Additionally, the data extracted may not even have a causal relationship, to begin with as mentioned earlier, resulting in more errors. Upon manual evaluation of the input, less than 20% of the causal sentences were identified correctly, and the performance drops further for uncommon cause words (accelerates, increases, contributes to, etc).

3.2 Approach II: Reinforcement Learning-based approach

With the emergence of transformer models and pre-trained language models, fitting a model on an annotated dataset becomes a trivial task, however, this may be infeasible considering such annotated data is not available. Hence, exploring unsupervised methods has an additional advantage - they open an avenue for a discussion where neatly annotated data is not a necessary blocker to get started with creating a valuable natural language asset. Many domains often require subject experts to annotate properly, and cannot be outsourced to popular annotation services.

Deep Reinforcement Learning in recent times has emerged as a promising approach that can utilize popular deep learning architectures, such as transformers, CNNs, LSTMs, etc., while also going a step further than function approximation toward generalized Artificial Intelligence. This is possible due to the way RL tasks are formulated as an optimization strategy, where we simulate an agent playing a finite sequential game to gradually improve the reward obtained at each step. The key difference is that this scalar reward neither needs ground truth labels nor has to be differentiable - as long as the reward magnitudes reflect the agent behaving favorably.

We propose an unsupervised framework for the causality extraction from sentences using the A2C or Actor Advantage Critic Method [7]. The advantage of this framework is that even though we lack ground-truth labels, essential for supervised learning, we can creatively use pre-trained models to assign a 'score' or evaluate our predictions. We use a combination of pre-trained models and hypotheses to formulate the score (μ_t). The basic idea behind this is, that if the predictions are correct, certain conditions must hold. For instance, if the predicted cause and effect are correct, firstly these entities should be noun phrases and secondly, if we frame a new sentence using these entities, this 'conclusion' sentence should be consistent or 'agree' with the premise sentence. We present the notations of our RL model in Table 1.

Reinforcement Learning Steps A typical RL problem consists of the following setup: a sequential task, where an *agent* starts at a initial position(s_0), and has to navigate through different *steps* to eventually reach an end point(s_T), which is referred to as completing an *episode*. At every step, the *agent* receives feedback on the decisions taken. Based on the feedback, at time t it tries to take an action(a_t) that will maximize the reward(r_t). Eventually, after multiple simulations of an *episode* and using an optimization algorithm,

Table 1. Notation for Approach II (refer to this for all the equations and formulations discussed in this approach

Notation	Description
t	Time step 't'
T	Maximum time steps in an episode.
υ	a random sub-sample of input sentences
θ_t	Given υ, represents predictions at time t
s_t	State at time step t(subject to definition)
a_t	Action taken at time step t. Action may not not be necessarily be the same as predictions, although they directly lead us to the prediction - for instance action can be something like softmax scores or probabilities, while θ_t are the actual textual prediction inferred from it.
μ_t	Given an υ and corresponding θ, this represents the score of the prediction or how good it is.
r_t	Represents reward at time step

the objective is to maximize the cumulative *reward* or $\sum_{t=0}^{T} r_t$ for an *episode*. We use this setup and define s_t, r_t and a_t to ensure that maximizing $\sum_{t=0}^{T} r_t$ will improve the prediction accuracy of labeling words in a sentence as *cause* and *effect*.

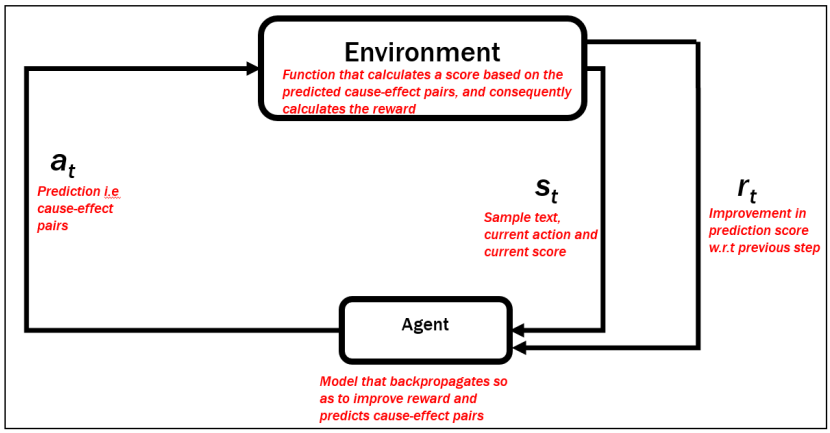

Fig. 6. Setup for using RL to extract cause-effect pairs

3.3 RL Task Description

For a particular episode, we pick a random subsample(υ) of sentences. At every step, the *agent*(in our case a neural network), takes s_t as an input and predicts a_t, also giving

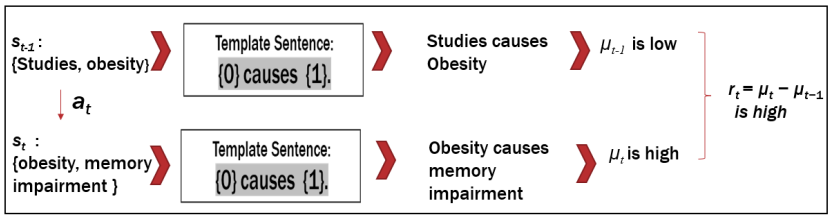

Fig. 7. μ_t i.e Scoring method explained

us θ_t. We score this prediction and assign value μ_t to it. Accordingly, since we want to use previous feedback and results to guide current action, we define the $state(s_t)$ as a collection of a time-invariant variable (input sentence) as well as two-time dependent variables (previous state and scores) incorporating the information of the trajectory after the start.

$$s_t = [\upsilon, .a_{t-1}, \mu_{t-1}]$$

Since RL algorithms optimize $\sum_{t=0}^{T} r_t$, we define our reward as

$$r_t = \mu_t - \mu_{t-1}$$

Thus, $\sum_{t=0}^{T} r_t$ is $\mu_t - \mu_0$, meaning optimizing cumulative reward is the same as improving the prediction score compared to random walk (based on our definition). Fig. 6 explains the RL steps and Fig. 7 the scoring method used in this approach.

As mentioned earlier, we can leverage RL algorithms for unsupervised learning since there are ways to use pre-trained models creatively in a way that allows us to automatically assign a score to a cause-effect prediction. Similar to how we built question 'templates' based on which cause words occurring in the sentence, we can build templates of 'conclusion' sentences for every cause word. For instance, for the cause word 'accelerates', the list of conclusion sentences are [0 accelerates 1.,1 is accelerated by 0., Acceleration of 1 is caused by 0].

Actor Advantage Critique Algorithm (A2C) The actor critique algorithm is based on Deep Q-learning Network (DQN) [8] algorithm. This RL framework is used along with actions and rewards designed based on our NLP tasks to extract cause and effect pairs. This network uses Value function and Q-values at each state to compute the usefulness and quality of the state. At each state s_t consisting of υ , a_{t-1} and μ_{t-1} where sentence stays constant where as a_{t-1} and μ_{t-1} are the feedback terms. The μ_{t-1} is a scalar output and a_{t-1} is a vector of $4\times$ $maxlen(\upsilon)$. We fix that the maximum length of a sentence is 80 words for our experimentation and we estimate the probability of every word to be a cause or an effect word. The output vectors for each word will have a size of four and each element will represent the probability of the word to be start of a cause phrase $\phi^s(\kappa)$, probability of the word to be end of a cause phrase $\phi^e(\kappa)$, probability of the word to be start of a effect phrase $\phi^s(\varepsilon)$, and probability of the word to be end of a effect

phrase $\phi^e(\varepsilon)$ respectively. Based on this probability distribution start and end indices of cause and effect phrases are determined.

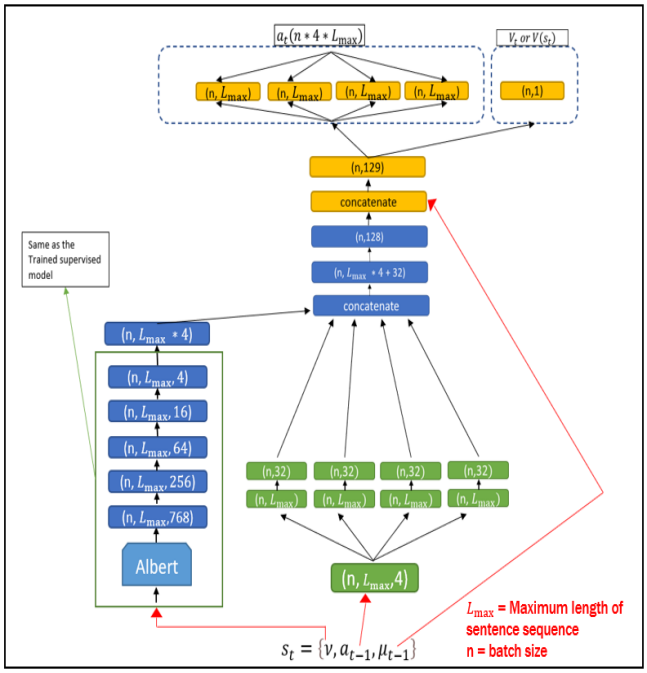

Fig. 8. Architecture for "Actor" portion of Approach II

Architecture and Setup The goal of our model to identify cause and effect phrases is shown in Figure 8 at each iteration of the state a sentence of length $len(d_i)$ is passed through Albert [10] a lighter version of BERT [9] based transformer model with 12 million parameters to generate sentence embeddings of size ($len(d_i)$,768). This output is then batch normalized [11] and is reduced by taking a mean across the length l resulting in vector of size (1,768). Then the action $a_t - 1$ output from previous state of size (80, 4) is reduced to (1,128) and batch normalized. This output $a_{t-1}^{|}$ and μ_{t-1} are combined to one single vector of size (1, 896), this output is further reduced and normalized to (1, 128) and combined with the scalar epsilon from previous state ε_{t-1}.

The RL approach seems promising from a theoretical perspective, in the sense that it eliminates the need for annotation, it does suffer from some practical issues. While we carried out training on the A2C model and managed to improve the reward by 30 percent, it did seem to saturate at that point. The reason is that our currently defined evaluation

function does not completely recreate the correct required evaluation, and suffers from error. Additionally, typically RL algorithms navigate in a more simplistic action space (discrete low dimensional action space), hence without 80×4 large dimensional as well as continuous action, the algorithm was not able to navigate properly[13]. While this can be fixed by altering the algorithm or using other variants of the Q learning base equation, it is outside the computational scope due to the sheer size of the Albert model parameters and our available resources.

4 Supervised Approaches

We will refer to the notations in Table 3 to formulate methodologies for both approaches III and IV

4.1 Annotated Dataset

After exploring unsupervised approaches, we move on to conventional supervised learning which would require annotated datasets. We used a combination of two datasets, each with annotation identifying multiple cause-effect entities in a dataset - the SemEval-2010 dataset [3] and the dataset used by [18]. The combined dataset had a total of 6832 unique sentences.

Challenges Investigating a few examples, we see that there is no recognizable pattern that links noun phrases and that exhibits a cause-effect relationship
Furthermore, there is a deep ambiguity when trying to understand the relationships even manually. Consider the following example - *the strong earthquake caused a blackout on the sound stage and short-circuited some of the neon-tubed violins.*
Here 'the strong earthquake' is the cause phrase, the effect phrase being either 'blackout' or 'neon-tubed violins'. Hence within a single sentence, there are multiple cause-effect pairs. Moreover, the collection of words 'blackout on the sound stage' can also be interpreted as one large noun phrase, which further adds to the ambiguity. This one-many characteristic, in particular, is important to note since it guided the design of our supervised approaches to a great extent.

From these examples, we learn that there does not exist a simplistic one-one mapping for the prediction, and simply picking a state-of-art transformer architecture like BERT[9] and fitting a supervised model will not work. Additionally, there seems to be a lot of these kinds of sentence structures, hence it cannot be altogether regarded as an outlier or a corner case. To summarize, the basic idea behind a Supervised approach is to predict cause effect as a direct sub-sequence of the input sentence sequence, using annotated cause-effect data as the basis for the same.

Data Pre-processing We implemented a rudimentary pre-processing on the annotated text such as the removal of non-alphanumeric characters and lower-casing. We then carefully extracted all the annotated pairs per sentence expanding the selection to get

data points where each point consists of a triplet of sentence, cause effect respectively. Due to the use of the transformer model and associated tokenizer, there is no need for stemming or lemmatization.
Pre-processing is important both before modeling and for extracting the noun phrases.

Noun Phrase (NP) extraction While comprehensive Noun Phrase (NP) extraction remains a complicated and ambiguous task, we attempt to capture as many noun phrases as possible, like other types of NPs' which are modeled to predict similar cause-effect pairs from a pool of candidates, rather than being deterministic for a given input sentence. We use a combination of noun chunk extraction and Spacy library's[36] noun phrase extraction to retrieve the Noun Phrases. Special care is taken to avoid repetition of the occurrence of two phrases in the noun phrases that mostly overlap each other. For instance, a noun phrase extractor will label both 'earthquake' and 'the earthquake' as possible noun phrases, however, we will omit the smaller one to avoid repetition.

4.2 Approach III : Using Siamese model for supervised prediction

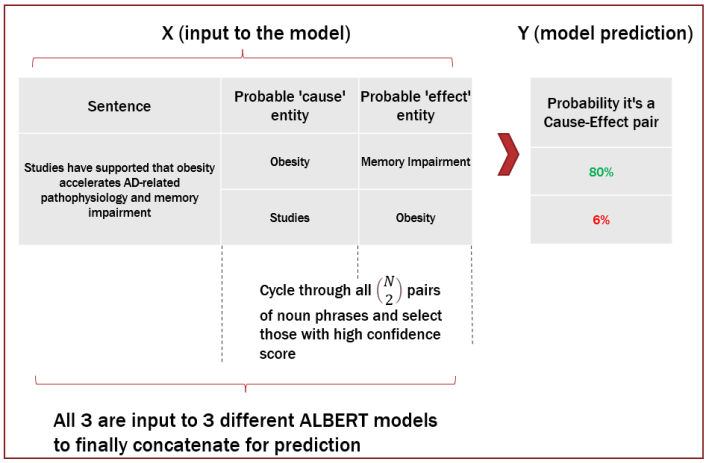

Fig. 9. Overview of the supervised Siamese approach

In this approach we use Siamese triplets for prediction. This means we use a model ξ to predict the probability

$$P_i^{\xi} = P(\theta_i^j = \hat{\theta}_i^j | xi = \{d_i, \hat{\theta}_i^j\})$$

where a triplet is $d_i, \kappa_i, \varepsilon_i$ (for the full list of notations used refer Table 2). Here, it is easy to populate a collection of probable solutions or sample solutions $\hat{\theta}_i$. We extract a list of

Table 2. Notation for Approach III and IV (refer to these notation chart for all the equations and formulations in these approaches)

Notation	Description
x_i	ith input data point in supervised prediction.
d_i	ith Document (Input sentence) . It is important to note that while in earlier works d_i would be the same as x_i, we are dealing with a one-to-many relation as far as document-entity pairs are concerned, hence x_i can be a combination of multiple entities as we shall see later.
η_i, N_i	Set and count of noun phrases respectively for sentence d_i
θ_i	Cause-effect entity solution set for d_i
$\theta_i^j = \{\kappa_i^j, \varepsilon_i^j\}$	One pair of correct solution for d_i, with $\{\kappa_i^j, \varepsilon_i^j\}$ being the jth cause and effect entities respectively.
$\phi^s(\kappa_i^j)$	Probability distribution for the 'start' index of κ_i^j. In vector form this will be an array of the same length as d_i, with each element representing the probability that κ_i^j starts at that position within d_i. Similary, $\phi^e(\kappa_i^j)$ stands for end index distribution, and $\phi^e(\varepsilon_i^j)$ for 'effect' entity probabilities. We typically estimate the predicted index by argmax(ϕ^s))
ξ	Neural Network extractor and prediction function, hence $\xi(x_i) = \theta_i$
P_i^{ξ}	Probability value of our prediction, ie, the confidence at which we can claim a predicted potential solution $\hat{\theta_i^j} \in \theta_i$. As we will dive further into methodologies later, we can formulate our outputs to predict this very probability.
$\hat{}$	Indicates a predicted or potential value, which might or might not be the ground truth value.
$len()$	Operator that return length of single dimensional vector input
M	Total number of labelled sentences.
ℓ	maximum length of sentence. As per experimentation, distribution of data and computational availability, this value is currently set to 40.

noun phrases or compound word entities from the sentence (we assume that the cause and effect entities will always belong to η_i), and then proceed to get all $\binom{N_i}{2}$ combinations of possible pairs from η_i. Based on each probable solution being accurate as per ground truth annotation or not, we accordingly assign positive and negative labels respectively. We then attempt to fit a binary classification model using start-of-art BERT architecture. During prediction, we extract those with a probability above a certain threshold. Fig. 9 presents an overview of the Siamese approach.

Experimentation revealed that the model did barely outperform a random walk prediction ($< 50\%$ correct predictions). We concluded that two possible reasons behind the low performance of this approach. Firstly, the formulation has a huge inherent class imbalance issue, since the number of negative triplets will be comparatively quite high. Secondly, we noticed that we need an architecture where each element in x_i has interaction with the other, and this interaction needs to be extracted by ξ, and not treated as individual features. This is expected since typical architectures while they calculate self-attention, they do not consider attention from amongst multiple separate text documents within a single input. This second reason behind the failure of this model is important to keep in mind moving ahead.

4.3 Approach IV : Using sentence-entity pair for supervised prediction

To address the issues faced earlier, both in terms of class imbalance as well as architecture, we utilize ξ to instead predict $\{\hat{\phi}^s(\varepsilon_i^j), \hat{\phi}^e(\varepsilon_i^j)\}$ with $x_i = \{d_i, \hat{\kappa_i^j}\}$. We thus indirectly predict the probability

$$P_i^{\xi} = P(\varepsilon_i^j = \hat{\varepsilon_i^j} | xi = \{d_i, \kappa_i^j\})$$

where $\hat{\varepsilon_i^j}$ is determined by the list of words (in order) starting with index $argmax(\hat{\phi}^s(\varepsilon_i^j))$ and ending with index $argmax(\hat{\phi}^e(\varepsilon_i^j))$ from d_i in vector form (for the full list of notations used refer Table 2). Hence, instead of populating a collection of probable pairs, we only populate a collection of probable 'cause' entities from the collection of noun phrases. We also utilize a slightly different Albert architecture which calculates relative attention weights from amongst the two separate entities in x_i and predicts the probability distribution of the start end indices directly, removing any imbalance issues. Please note that keeping the same architecture, we can alternatively model with $x_i = \{d_i, \hat{\varepsilon_i^j}\}$ as well, with the same methodology and architecture. The choice to predict effect and train on sentence + cause was based on experimentation and subsequent results. It is also important to note that by predicting indices, and not using a seq-to-seq or generative model, we turn this into a deterministic exercise. This way, we have a shorter search space and we can have better predictions. This was also evident from our experiments using alternative methods.

Architecture ALBERT [10] is a transformer architecture based on BERT [9] but with much fewer parameters. It achieves this through two parameter reduction techniques. The first is a factorized embeddings parameterization. By decomposing the large vocabulary embedding matrix into two small matrices, the size of the hidden layers is separated

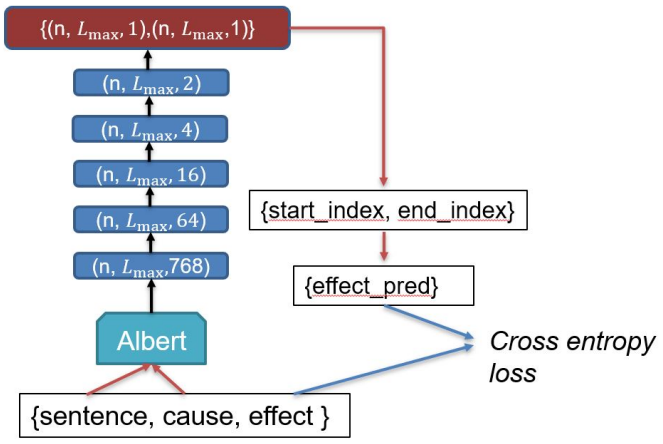

Fig. 10. Architecture of Approach IV

from the size of the vocabulary embedding. This makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings. The second technique is cross-layer parameter sharing. This technique prevents the parameter from growing with the depth of the network. Since we received a sequential output, instead of reducing the dimension directly (taking mean or sum), we reduce it by using linear neural networks before reducing the size to that of the output (refer to Fig. 10). The particular Albert model in question takes in a pair of sentences as input, this way attention between the two sentences (first being the main input sentence and the second being the cause entity), is also taken into consideration and is the key contributor to achieving higher performance.

Training Consider the input sentence d_i = '*the distraction caused by the students, coupled with limited vision down the track, caused the incident to occur.*' (refer to Table 3). As per annotation, len(θ_i) = 3, meaning we have 3 distinct cause-effect pairs which are θ_i^0 = {'limited vision' ,'the incident'}, θ_i^1 = {'the students','the distraction'} θ_i^2 = {'the distraction','the incident'}. This would mean the total number of data points are $\sum_{i=0}^{M} len(\theta_i)$

- **Tokenization**: We tokenize pairs of sentences and cause entities. Tokenization consists of assigning a finite integer token for every unique word, punctuation as well special tokens like 'sentence start', 'sentence end', and 'padding' tokens. The start and end tokens tell the model to not consider anything other than the the part of sequence between these two tokens when calculating the loss function. Hence post tokenization, $len(d_i) = \ell \; \forall i$
- **Training convergence**: As the output for prediction is a multi-dimensional categorical variable (start and end index), it was trained until convergence of the validation cross entropy loss. It should be noted that there can exist multiple data points per

sentence, since a sentence has more than one cause-effect pair. To prevent data leakage, we took care to make sure that

– **Output probability distribution**: As mentioned before, the model is trained for multi-class classification, the output being of dimension $2 * \ell$, each dimension representing vectors $\phi^s(\varepsilon_i^j)$ and $\phi^e(\varepsilon_i^j)$ respectively. Consequently the probability P_i^{ξ} will be formulated by

$$P(\varepsilon_i^j = \hat{\varepsilon_i^j} | xi = \{d_i, \kappa_i^j\}) = max(\phi^s(\varepsilon_i^j)) * max(\phi^e(\varepsilon_i^j))$$

It is important to note that due to the nature of the dataset, while the training data utilizes both d_i and κ_i^j, the prediction part would only have d_i as input, hence looking at the probability distribution is important as we formulate the prediction process. We observed from the distribution that majority of the distribution lies below approximately 0.8.

Prediction

– **Creating Noun Phrase set**: We have already covered how we extract noun phrases, furthermore it should be noted that there is an assumption to be made that all potential cause phrases belong to this set of noun phrases, ie, $\kappa_i^j \in N_i \forall i..$ Hence during prediction, we take the input sentences and create test data points such that we have N_i data points, where $x_i = \{d_i, \hat{\kappa_i^j}\}$ and $\hat{\kappa_i^j}$ is basically an iteration over every element in η_i. This also means the predicted number of cause-effect pairs, as per our assumption will be lesser than the number of noun phrases, ie $len(\hat{\theta}_i) \leq N_i$. For instance, for the example sentence mentioned above, the set of noun phrases are {'the distraction', 'the students', 'limited vision', 'the track', 'the incident'}. An important point to note here is that, during evaluation, we take special care to make sure that even partially predicted phrase is accepted as long as there is a significant overlap. This means if the ground truth phrase is 'limited vision' and our network predicts 'vision', it is considered acceptable or correct prediction, and the rule holds vice versa too. Similarly, we also compensate for missing or additional articles, hence 'the students' is equivalent to 'students'.
– **Prediction**: We predict $\hat{\phi^s(\varepsilon_i^j)}$ and $\hat{\phi^e(\varepsilon_i^j)}$ using the trained ξ giving us potential N_i pairs for every sentence, which we filter down based on probability cutoff as mentioned in the training process, giving us the final list of predictions.

5 Evaluation

5.1 Evaluation Plan and strategy

The four approaches presented in the previous section provided the inference and insights for the next. Beginning with Approach I, we learn that although transformer-based pre-trained models can learn complicated natural language information, they still do not quite have sufficient task-specific understanding to be used as-is for our task. However, they do give a fair comparable insight, that is, provided two different sets of predictions,

the confidence score or loss returned by a pre-trained model does inform us which set of predictions is better. This formed our intuition for designing the reward function for Approach II. Solving RL problems, however, requires a huge amount of experimentation and computational demand. Not only does it require exponentially more amount of training simulations (each involving multiple epochs), it also requires an extensive search of the hyper-parameter space which was beyond our scope in terms of building a robust, cost-efficient, and deployment-ready solution. However, there is promise in the experiment design and formulation of the RL approach, and there is scope to pursue this as an engineering challenge. All the experiments were was conducted on a variable RAM (ranging from 16GB to 64GB) virtual machine with a single Tesla P100 or V100 GPU. Fig. 11 presents the memory and computing resources consumed during the training process.

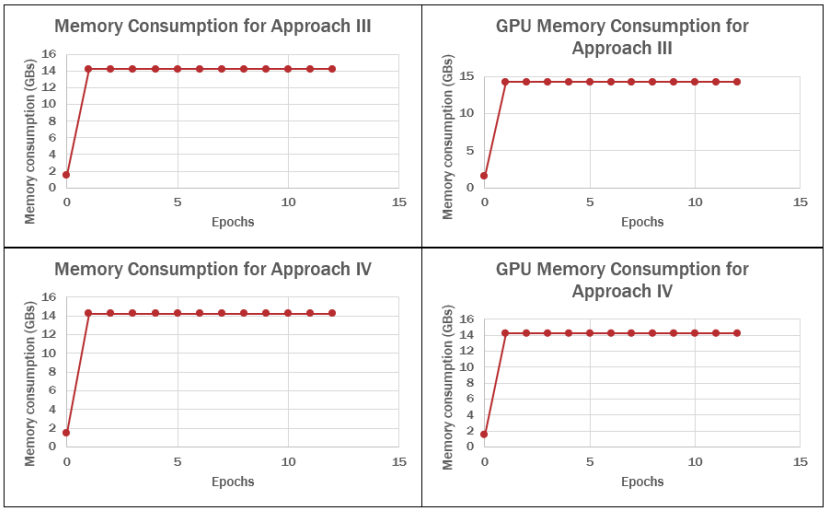

Fig. 11. Overview of memory consumption for CPU and GPU

As we move on to Supervised approaches, the nuances and challenges of the dataset need to be kept in mind as we explore this further. As discussed before, a sentence can have more than cause-effect pair, hence rather than having each data point just one sentence, each data point is a unique collection of (sentence, cause, effect).
Once again, based on our assumption that cause-effect entities will belong to the larger set of noun phrases from a sentence, we can cycle through all the combinations and see if we can train a model to predict the correct ones. This was Approach III - a binary classification model which predicted if this collection or triplet represented a correct cause-effect relationship. The binary labels allowed for simplistic logistic loss and standard statistical performance metrics of Precision, Recall, and F1. However, not only did it have a class imbalance (since only 2 or 3 out of $\binom{N}{2}$ combinations will be 'True' label, rest will be 'False' labels for every sentence), it also looked at the 3 textual

entities as independent features in the Neural Network. This means although the neural network is capable of extracting information from *within* the entities, it is difficult to detect the relationship *amongst* them.

To resolve both the problems, the final architecture and approach i.e approach IV was formulated which treated pairs of the sentence and one entity as input sequence - this means that it is ultimately treated as a single input and allows for self-attention between sentence and the causal entity. If the input is the sentence and the cause phrase, the output will be the effect phrase, represented by its start and end word index. Predicting the output phrase using indices rather than approximating the output text based on predicted embedding is an important decision as well because it means a much smaller search space. For the neural network loss function, a logistic loss is calculated for both the start and end index separately and added for training. Both predicted and actual entities are reflected as sequences (words from a start index to end index) and we find the length of intersection of these sequences. Accordingly, we use the following equations for precision, recall, and F1 Score for every data point in the validation set and then average

$$Precision = \frac{\text{length of intersection}}{\text{length of predicted entity}}$$

$$Recall = \frac{\text{length of intersections}}{\text{length of actual entity}}$$

$$\text{F1 Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

For both the supervised approaches, training was done using the Adam optimization algorithm with early stopping if validation loss does not reduce more than 5%. The Python-based Optuna Framework [37] was used for tuning to see if different hyperparameters (learning rate, batch size, dropout, etc.) yielded better results however it resulted in less than 5% in performance which was also around the variance found in running the models on different random seed, hence can be attributed to noise. Due to the dataset not being exceedingly vast, a hyper-parameter search was not necessarily expected to give improved results.

Table 3. Description of the training and validation setup for Approach III and IV

Measure	**Approach III**	**Approach IV**
Size of training dataset	11284	5184
Size of validation dataset	4710	2211
Loss function	Binary cross entropy	Cross Entropy

5.2 Results

In the following section, we present the detailed results from all of our approaches.

Unsupervised methods - Approach I and II Evaluating unsupervised methods, in particular, is a challenge considering that we have considered it as a complete end-to-end unsupervised process, that is, no labeled dataset was used from formulation to evaluation. Nonetheless, we have some qualitative evaluations from each approach that informed the previous approaches. It was obvious from the results of Approach I that pre-trained models tend to perform exceedingly well the more similar the domain of the dataset and characteristics are to the original dataset it was originally trained upon. For instance, DistillBERT (which is the parameter reduced version of BERT) performs well with shorter sentences but exceedingly worse with longer sentences, and vice-versa applies to BERT. Pre-trained models also couldn't perform well with large technical words or especially with sentences with pairs of cause-effect relationships in the sentence. Hence although this approach failed, it gave an important conclusion which is that modern supervised algorithms although achieve near-human accuracy, their learning is more task-driven and not generalized understanding of the English language and grammar. The same applies to Approach II, where although we were successful in increasing the reward to a level of 30% improvement before convergence, reward optimization did not necessarily result incorrectly extracted cause-effect pairs, as the reward is still based on pre-trained models.

5.3 Supervised methods - Approach III and IV

Although previous works achieve an appreciable level of performance, the assumptions and over-simplifications involved, such as only one unique pair of cause-effect per sentence, the entities limited to 1-2 words, non-overlapping noun phrases, etc make them less viable in practical application. Our work makes few assumptions and caters more comprehensively to so-called edge cases. The setup for these two approaches is described in Table 3 and their performance is shown in Table 4.

Table 4. Comparison between the Supervised methods - Approach III and IV. It is evident that although both methods utilize the same dataset and a transformer based architecture, Approach IV is pretty superior in performance. The performance and learnings from Approach III informed our key modifications in the architecture. Our final approach's performance gives a conclusive ground for proof of concept of the architecture and training setup.

Metric	**Approach III**	**Approach IV**
Precision	47	70
Recall	53	80
F1 Score	50	73

6 Conclusion

Cause-Effect extraction is a very nuanced and complex exercise that not only requires grammatical understanding but also data-specific considerations for any model to predict.

Using pre-trained models directly or indirectly through reinforcement learning has huge potential for unsupervised extraction – and this approach can be expanded and repurposed to multiple other tasks. However, reinforcement learning also requires clearly defined and quantifiable reward functions, as well as comparatively superior computational capacities to complete experiments. In the case of Supervised models, we were able to handle the complication of multiple pairs within a sentence as well as build an architecture that can learn and converge successfully. Although this method outperforms all the others, it still depends on investment in procuring annotated (labeled data), and also the utility is limited to the same or slightly different tasks.

7 Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1948322.

References

1. Tran, N., Ceroni, A., Kanhabua, N. & Niederée, C. Back to the past: Supporting interpretations of forgotten stories by time-aware re-contextualization. *Proceedings Of The Eighth ACM International Conference On Web Search And Data Mining*. pp. 339-348 (2015)
2. Buttcher, S., Clarke, C. & Cormack, G. Information retrieval: Implementing and evaluating search engines. (Mit Press,2016)
3. Hendrickx, I., Kim, S., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L. & Szpakowicz, S. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. *Proceedings Of The 5th International Workshop On Semantic Evaluation*. pp. 33-38 (2010,7), https://aclanthology.org/S10-1006
4. Gurulingappa, H., Rajput, A., Roberts, A., Fluck, J., Hofmann-Apitius, M. & Toldo, L. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal Of Biomedical Informatics*. **45**, 885-892 (2012), https://www.sciencedirect.com/science/article/pii/S1532046412000615, Text Mining and Natural Language Processing in Pharmacogenomics
5. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv Preprint ArXiv:1910.01108*. (2019)
6. Rajpurkar, P., Zhang, J., Lopyrev, K. & Liang, P. Squad: 100,000+ questions for machine comprehension of text. *ArXiv Preprint ArXiv:1606.05250*. (2016)
7. Mnih, V., Badia, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. & Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. (2016)
8. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M. Playing Atari with Deep Reinforcement Learning. (2013)
9. Devlin, J., Chang, M., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Preprint ArXiv:1810.04805*. (2018)
10. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. & Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *ArXiv Preprint ArXiv:1909.11942*. (2019)
11. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*. **abs/1502.03167** (2015), http://arxiv.org/abs/1502.03167
12. Yang, J., Han, S. & Poon, J. A survey on extraction of causal relations from natural language text. *Knowledge And Information Systems*. (2022,3), https://doi.org/10.1007/s10115-022-01665-w

13. Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions On Neural Networks*. **5**, 157-166 (1994)
14. Sorgente, A., Vettigli, G. & Mele, F. Automatic Extraction of Cause-Effect Relations in Natural Language Text. *DART@AI*IA*. (2013)
15. Karimi, S., Wang, C., Metke-Jimenez, A., Gaire, R. & Paris, C. Text and Data Mining Techniques in Adverse Drug Reaction Detection. *ACM Computing Surveys*. **47** pp. Article 56 (2015,3)
16. Egami, N., Fong, C., Grimmer, J., Roberts, M. & Stewart, B. How to Make Causal Inferences Using Texts. (2018)
17. Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J. & Kidawara, Y. Toward Future Scenario Generation: Extracting Event Causality Exploiting Semantic Relation, Context, and Association Features. *ACL*. (2014)
18. Li, Z., Li, Q., Zou, X. & Ren, J. Causality extraction based on self-attentive BiLSTM-CRF with transferred embeddings. *Neurocomputing*. **423** pp. 207 - 219 (2021), http://www.sciencedirect.com/science/article/pii/S0925231220316027
19. Sharp, R., Surdeanu, M., Jansen, P., Clark, P. & Hammond, M. Creating Causal Embeddings for Question Answering with Minimal Supervision. *Proceedings Of The 2016 Conference On Empirical Methods In Natural Language Processing*. pp. 138-148 (2016,11), https://www.aclweb.org/anthology/D16-1014
20. Müller, R. & Huettemann, S. Extracting Causal Claims from Information Systems Papers with Natural Language Processing for Theory Ontology Learning. *HICSS*. (2018)
21. Sasaki, S., Takase, S., Inoue, N., Okazaki, N. & Inui, K. Handling Multiword Expressions in Causality Estimation. *IWCS 2017 — 12th International Conference On Computational Semantics — Short Papers*. (2017), https://www.aclweb.org/anthology/W17-6937
22. Zhao, S., Jiang, M., Liu, M., Qin, B. & Liu, T. CausalTriad: Toward Pseudo Causal Relation Discovery and Hypotheses Generation from Medical Text Data. *Proceedings Of The 2018 ACM International Conference On Bioinformatics, Computational Biology, And Health Informatics*. (2018)
23. Kang, D., Gangal, V., Lu, A., Chen, Z. & Hovy, E. Detecting and Explaining Causes From Text For a Time Series Event. *Proceedings Of The 2017 Conference On Empirical Methods In Natural Language Processing*. pp. 2758-2767 (2017,9), https://www.aclweb.org/anthology/D17-1292
24. Girju, R. & Moldovan, D. Text Mining for Causal Relations. *FLAIRS Conference*. (2002)
25. Chang, D. & Choi, K. Causal Relation Extraction Using Cue Phrase and Lexical Pair Probabilities. *IJCNLP*. (2004)
26. Mostafazadeh, N., Grealish, A., Chambers, N., Allen, J. & Vanderwende, L. CaTeRS: Causal and Temporal Relation Scheme for Semantic Annotation of Event Structures. *EVENTS@HLT-NAACL*. (2016)
27. Lee, D. & Shin, H. Disease causality extraction based on lexical semantics and document-clause frequency from biomedical literature. *BMC Medical Informatics And Decision Making*. **17** (2017)
28. Qin, P., Xu, W. & Wang, W. Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning. *Proceedings Of The 56th Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers)*. pp. 2137-2147 (2018,7), https://www.aclweb.org/anthology/P18-1199
29. Ho, S. Causal Learning versus Reinforcement Learning for Knowledge Learning and Problem Solving. *AAAI Workshops*. (2017), http://aaai.org/ocs/index.php/WS/AAAIW17/paper/view/15182
30. Marcheggiani, D. & Titov, I. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. *CoRR*. **abs/1703.04826** (2017), http://arxiv.org/abs/1703.04826

31. Asghar, N. Automatic Extraction of Causal Relations from Natural Language Texts: A Comprehensive Survey. *CoRR*. **abs/1605.07895** (2016), http://arxiv.org/abs/1605.07895
32. Keith, K., Jensen, D. & O'Connor, B. Text and Causal Inference: A Review of Using Text to Remove Confounding from Causal Estimates. *CoRR*. **abs/2005.00649** (2020), https://arxiv.org/abs/2005.00649
33. Narasimhan, K., Yala, A. & Barzilay, R. Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning. *CoRR*. **abs/1603.07954** (2016), http://arxiv.org/abs/1603.07954
34. Eichler, M. & Didelez, V. Causal Reasoning in Graphical Time Series Models. *Proceedings Of The Twenty-Third Conference On Uncertainty In Artificial Intelligence*. pp. 109-116 (2007)
35. Ittoo, A. & Bouma, G. Extracting Explicit and Implicit Causal Relations from Sparse, Domain-Specific Texts. *NLDB*. (2011)
36. Honnibal, M., Montani, I., Van Landeghem, S. & Boyd, A. spaCy: Industrial-strength Natural Language Processing in Python. (Zenodo,2020), https://doi.org/10.5281/zenodo.1212303
37. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings Of The 25rd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. (2019)