NEURAL MODELS FOR CAUSAL INFORMATION EXTRACTION USING DOMAIN ADAPTATION

Anik Saha

Submitted in Partial Fullfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Approved by:
Dr. Bülent Yener, Chair
Dr. Ali Tajer
Dr. Tianyi Chen

Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, New York

[August 2023]
Submitted June 2023

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to express my gratitude to my advisor, Prof. Bülent Yener, for his guidance and professional insight in steering my research and helping me to critically think about any problem. I am grateful to Prof. Alex Gittens for sharing his knowledge and expertise in the rapidly changing field of machine learning and guiding my research. His consistent support and invaluable advice has profoundly influenced my research.

I would like to thank the members of my thesis committee, Prof. Ali Tajer and Prof. Tianyi Chen, for their insightful feedback and valuable comments that guided the direction of my thesis.

I am grateful to Dr. Oktie Hassanzadeh for mentoring me during my externship at IBM Research and providing his guidance on my research. I would like to express my appreciation to Dr. Kavitha Srinivas and Dr. Jian Ni from IBM Research for their professional knowledge and expert advice.

I want to thank the RPI Department of Electrical, Computer and Systems Engineering for the continued academic and financial support. I would also like to acknowledge the RPI Cognitive and Immersive Systems Lab for their support during my graduate study.

ABSTRACT

The task of identifying causality related events or actions from text is an important step in building knowledge graph of events and consequences from the vast amount of unlabeled documents available to us in the digital age. As there has been very little work on this problem, we perform a comparative study on different neural models on 4 different data sets with causal labels. We train sequence tagging and span based models to extract causally related events from their textual description. Our experiments affirm the fact that large pre-trained language models, i.e. BERT, can be fine-tuned on labeled data sets to outperform traditional deep learning models like LSTM. Our results show that span based models are better at classifying spans of words as cause or effect compared to sequence tagging models using the same pre-trained weights from BERT. The length of the spans labeled as causes and effects in a data set also has a significant impact on the advantage of using a span based model.

Towards the goal of developing a general purpose model for extraction of causal knowledge from text, we focus on the unsupervised domain adaptation (UDA) scenario where we adapt a model trained on a source domain to a new domain without any label. Several studies on the UDA task for text classification have shown the effectiveness of the adversarial domain adaptation method. We investigate the effect of integrating linguistic information in the adversarial domain adaptation framework for the causal information extraction task. We show the advantage of leveraging the word dependecy relationship in adapting word based neural models like LSTM to new domains. We also find that guiding the adversarial domain classifier to adapt the specific task classifier output is more effective than requiring the encoder model outputs to have similar distributions in two domains.

CHAPTER 1
INTRODUCTION

Causal information extraction is the task of automatically identifying causal information from text by extracting pairs of words or phrases representing causes and effects. By extracting causal information from a large unlabeled corpus, we can build a knowledge graph of events and consequences for any domain. Although a large number of papers have been published on extracting causal information from text, most of them have focused on detecting the existence of a causal relationship [1],[2]. Prior works have explored methods such as detecting whether a causal explanation exists between two events [3], incorporating external knowledge from ConceptNet [4] to augment event causality identification [5]. While this approach helps us identify causal sentences from unlabeld corpora, it does not provide causally related entities or events.

We are interested in the problem of identifying cause and effect spans as this is an important step for automating knowledge graph construction and event forecasting. For example, in the sentence: "A second problem is *the persistent frustation of false alarms*, which can make *urban police less than enthusiastic about responding to small businesses.*" we want to extract the cause *the persistent frustation of false alarms* and the effect *urban police less than enthusiastic about responding to small businesses*. This problem is still underexplored in the literature except the work from [6] where they trained separate models for causality identification and cause-effect extraction with adversarial domain adaptation. They train a sequence tagging model to predict the cause and effect spans in a sentence. We experiment with sequence tagging models and span based models based on large pre-trained language models like BERT for the causal span prediction task. Both types of models based on BERT pre-trained weights beat the graph convolution network (GCN) based model on multiple data sets with causality labels.

Since there are very few data sets available with cause-effect labels and most of them have small number of samples, it is important to quantify the usefulness of domain generalization methods for the causality extraction task. Pre-trained language models like BERT perform very well when fine tuned on a labeled data set. But for data sets or domains lacking labeled sentences, we can use unsupervised domain adaptation methods to build an effective causality extractor model. Although unsupervised domain adaptation (UDA) is a well

studied problem in computer vision problems, most UDA papers on nlp applications have focused on text classification. Unlike image data sets, we have access to traditional liguistic parsers to automatically extract linguistic information like Parts-of-speech tags, dependency relationships etc. Hence, we investigate the importance of such liguistic information in helping adversarial domain adaptation methods [7] perform better on tagging causes and effects in a sentence.

1.1 Contributions

We studied the performance of different causality extraction models through experiments on multiple labeled data sets. Here we list our Contributions

- We compare sequence tagging models and span based models by their performance on 4 labeled data sets and establish state of the art scores on all data sets
- We show the advantage of utilizing linguistic word relations in the unsupervised domain adaptation of a neural model for the first time to the best of our knowledge
- We demonstrate the importance of focusing on the task classifier output instead of the sentence embedding for the adversarial domain adaptation framework

In chapter 2, we report the performance of sequence tagging and span based models for the cause-effect extraction task on four labeled data sets. We focus on the domain adaptation performance of sequence tagging models in the adversarial domain adaptation framework using one source data set and three target data sets in chapter 3.

CHAPTER 2
A CROSS-DOMAIN EVALUATION OF APPROACHES FOR CAUSAL KNOWLEDGE EXTRACTION

Although the causal knowledge extraction task is important for language understanding and knowledge discovery, recent works in this domain have largely focused on binary classification of a text segment as causal or non-causal. In this regard, we perform a thorough analysis of three sequence tagging models for causal knowledge extraction and compare it with a span based approach to causality extraction. Our experiments show that embeddings from pre-trained language models (e.g. BERT) provide a significant performance boost on this task compared to previous state-of-the-art models with complex architectures. We observe that span based models perform better than simple sequence tagging models based on BERT across all 4 data sets from diverse domains with different types of cause-effect phrases.

2.1 Introduction

Automatic extraction of causal knowledge is an important task for language understanding. The causal relationship between entities and events can be used to build a forecasting model for future events. Causality information is expressed in many different forms in natural language. There might be explicit indicators like *because*, *therefore*, *result in* etc. A sentence can also encode causal relationship without using any causal connective word. So rule based methods are limited in their ability to extract causal knowledge from such texts. For a causality extraction system, it is important to understand the semantics of the text and recognize the pattern of sentences containing causal knowledge. The flexibility of neural models to represent any pattern in data makes them a strong candidate for this task.

To extract causal relationship in a text segment, the first step is to determine the existence of causal relationship. The next step is extracting the related cause and effect entities. Figure 2.1 shows some examples of causes and effects in the data sets used in experiments for this work. In recent years, there has been an increasing interest in developing neural models for causality extraction. These models focus on solving the binary problem of detecting if a sequence of words contains causal information. But we are more interested in extracting the related cause and effect pair from a causal text segment. Surprisingly, there is a scarcity of studies on performing this important knowledge extraction task both in terms

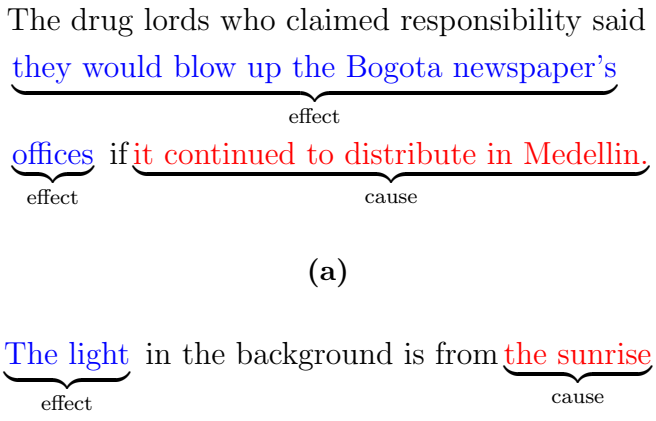

Figure 2.1: Example causal sentences from (a) BeCauSE data set and (b) SemEval-2010 data set

of models and data sets. In this work, we assemble data sets with cause and effect span labels from different domains and convert them to a standard format [8][1]. Then we evaluate the performance of four neural models and analyse the results to determine the effect of the model classes and different data set attributes. We also study two modes of evaluation for the span extraction or sequence labeling tasks.

The main contribution of the work is to compare three recent state-of-the-art models for cause and effect extraction across multiple domains, to elucidate the importance of: (i) differences in the data sets/domains, (ii) architectures of the models, and (iii) evaluation metrics towards their performance. Because different papers introduced and evaluated different models within a specific domain, we need a direct comparison of models across domains to understand these factors. We apply a span-based relation extraction model to the causal knowledge extraction task, and show it consistently outperforms these baselines in terms of the most demanding accuracy metric when maximum span length is tuned well for the domain. Large variance in performance of models across domains point towards the need to explicitly evaluate each model across domains when they are introduced, as well as the need to design models explicitly to work well across domains.

[1]https://github.com/phosseini/CREST

2.2 Related Work

Research on causal relation extraction have been mostly focused on detecting the existence of causality. Prior works can be broadly categorized into 3 classes based on their primary objective; text classification, causal relation identification and cause-effect extraction / sequence labelling [9].

Text Classification In text classification, the task is to determine if a text segment contains causal information. The model is not required to extract the cause or effect entities in the text. Features were manually selected in [10] to train a bagging tree for classifying sentences as causal or not causal. The features were based on the connector words, their modifiers, the class and tense of the cause and effect verbs. A linear SVM model was trained in [1] using features created from parallel corpus and lexical features from WordNet, FrameNet, VerbNet. They also created a data set for causal sentence classification using the parallel articles from Simple and English Wikipedia.

Causal Relation Identification Here, the task is to judge whether a pair of entities or events have a causal relationship given a context, i.e. a sentence. This is also a classification problem approached by several recent papers using deep learning methods. A multi-column CNN model was trained in [11] with 8 columns to determine if a causality relationship exist between two event mentions. They used 5 columns to feed vector representation of the event mentions and different fragments of the context. The other 3 columns take in background knowledge from different sources. A restricted hidden naive bayes (RHNB) model was proposed in [12] to extract candidate phrases from sentences and determine their relationship. A knowledge-oriented CNN model was trained for causality extraction in [13] using background knowledge from WordNet and FrameNet . There is a knowledge-oriented channel and a data-oriented channel in the model to process lexical knowledge and the input sentence respectively. The event causality identification task was addressed in [5] using the pre-trained language model, BERT as an encoder. They extracted knowledge from ConceptNet to augment the input sentence to the model. A masking loss was also used to predict the event mentions in text.

Cause-effect Extraction In this task the goal is to find the span of the cause and effect entities or events in a text segment if there exists a causal relationship. We are interested in this task as it helps us utilize the vast amount of text available online to 'discover' causal relationships. A liguistically informed Bidirectional LSTM model was developed in [14] by annotating the words in the SemEval-2010 [15] data set with cause, effect and causal connective labels. They evaluated their performance on cause, effect and causal connective word extraction separately. Also, [16] separately annotated the SemEval-2010 data set and trained a self-attentive BiLSTM-CRF model with Flair word embeddings. They addressed the problem of multiple causality where one cause can be related to more than one effect. A sequence tagging model was developed in [6] using a graph convolution network (GCN) added to the output of a bidirectional LSTM based on the dependency relationship between words. They created a data set by labeling medical articles from Wikipedia and proposed a domain adaptation technique using adversarial training method.

2.3 Model Description

We train 4 different models on labeled causal relation data sets. They are broadly divided into two classes: sequence tagging model and span based model.

2.3.1 Sequence Tagging Models

Here, the cause-effect extraction task is modeled as a sequence tagging task. The model predicts a tag for each word in the input sentence or paragraph. We use tagging schemes like Named Entity Recognition (NER) to train these models.

2.3.1.1 SCITE

Flair-BiLSTM-CRF This is a bidirectional LSTM model with a CRF output layer from [16]. They use pre-trained Flair [17] embeddings and pre-trained word embeddings from [18]. A character level convolutional network is used to build word embeddings from character embeddings. The embedding for each word is a concatentation of these 3 types of embeddings.

$$E_w = [E_c; E_F; E_d] \tag{2.1}$$

where E_c is the word embedding from character level embeddings, E_F is the pre-trained Flair embedding and E_d is the dependency based word embedding trained in [18].

The bidirectional LSTM has a forward and a backward LSTM layer that takes in the word embeddings E_w. The output embeddings from forward and backward LSTM layers are concateneted to obtain the output embeddings. Next, they add a multi-headed self-attention layer over these embeddings similar to a transformer model. Finally, there is a CRF layer to model the sequential relation of words in representing causality. A BIO tagging scheme is used to label the cause and effect spans in the data set. The model is trained to maximize the log likelihood of the correct label sequence.

2.3.1.2 GCE

BiLSTM-GCN We use the graph convolution encoder (GCE) model from [6]. Here, the graph for a sentence is built using dependency parsing relations. In the adjacency matrix of the graph, the corresponding entry of two tokens related according to dependency parse has a value of 1 and all other values are zero. A graph convolution network uses the graph structure based on this adjacency matrix to obtain embeddings for words (nodes). The input to the GCN is a sequence of embeddings from a bidirectional LSTM built on top of word embeddings.

$$\mathbf{h}_{\text{GCN}}(X) = f^{\text{pool}}(\text{GCN}(\text{BiLSTM}(\text{X}))) \tag{2.2}$$

where, X is the seqence of input word embeddings and f^{pool} is the pooling function to obtain an embedding for the sentence. They also concatenate the BiLSTM output with the GCN output.

$$f_{enc}(X) = \text{FFNN}(\mathbf{h}_{\text{GCN}}(X); \mathbf{h}_{\text{BiLSTM}}(X)) \tag{2.3}$$

where FFNN is a feed-forward network.

The output from the encoder is fed to a classifier layer for cause and effect extraction. This model is trained to predict IOBES style tags for each word in the input.

2.3.1.3 BERT Sequence Tagger

BERT-base This model also uses the IOBES tagging scheme. We use the pretrained Bert-base [19] model to obtain embeddings for the sentence. A softmax layer converts the output embeddings to label predictions.

$$y_i = \text{Softmax}(W.e_i + b) \tag{2.4}$$

where, e_i is the BERT embedding for the i-th token and y_i is the softmax output.
This model (Fig. 2.2) is fine-tuned on the annotated data set to extract cause and effect spans.

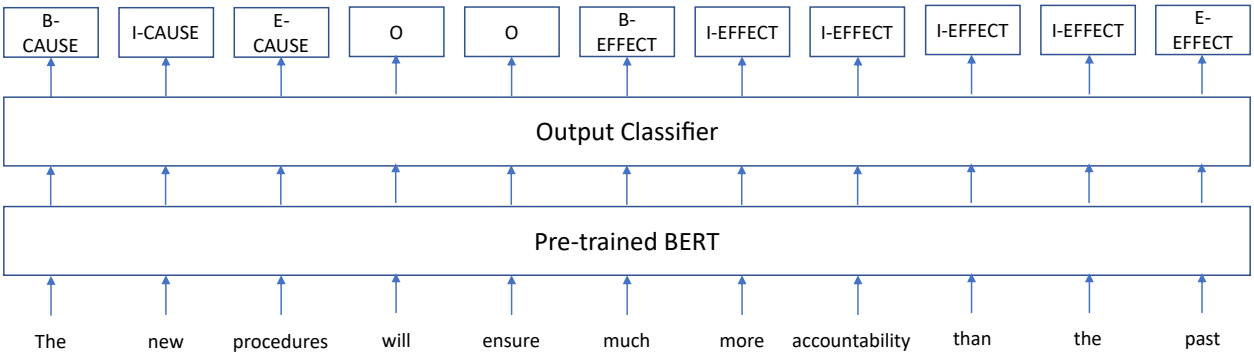

Figure 2.2: BERT sequence tagger

2.3.2 Span Based Model

SpERT In this model [20], the spans of causes and effects are labeled along with the related pairs. The model uses the output embeddings from BERT to classify candidate spans and their relations. (Fig. 2.3). The model classifies a list of candidate spans generated by taking all possible sequences of words upto a maximum length from the sentence. The span representation is formed by combining the BERT embeddings of all tokens in the sentence. A width embedding matrix is also trained to represent the size of the candidate span with a fixed size embedding. Span embedding is the concatenation of the max-pooled token embeddings in the span, the span size embedding and the [CLS] embedding from BERT.

$$\mathbf{e}(s) = f(\mathbf{e}_i, \mathbf{e}_{i+1}, \ldots \mathbf{e}_{i+k}) \circ w_{k+1} \circ c \tag{2.5}$$

,where $\mathbf{e}(s)$ is the span embedding, $\mathbf{e}_i$ the embedding for i-th token and w is the width embedding, c is the CLS token embedding. A span classifier labels the candidate spans as cause or effect or other using span embeddings.

$$y_s = \text{softmax}(W_s.\mathbf{e}(s) + b_s) \tag{2.6}$$

Then a relation classifier layer predicts the related pairs of spans out of all possible pairs. A pair of spans is represented by the concatenation of the output embeddings from the span

classifier and the max-pooled embeddings of the tokens in between the spans.

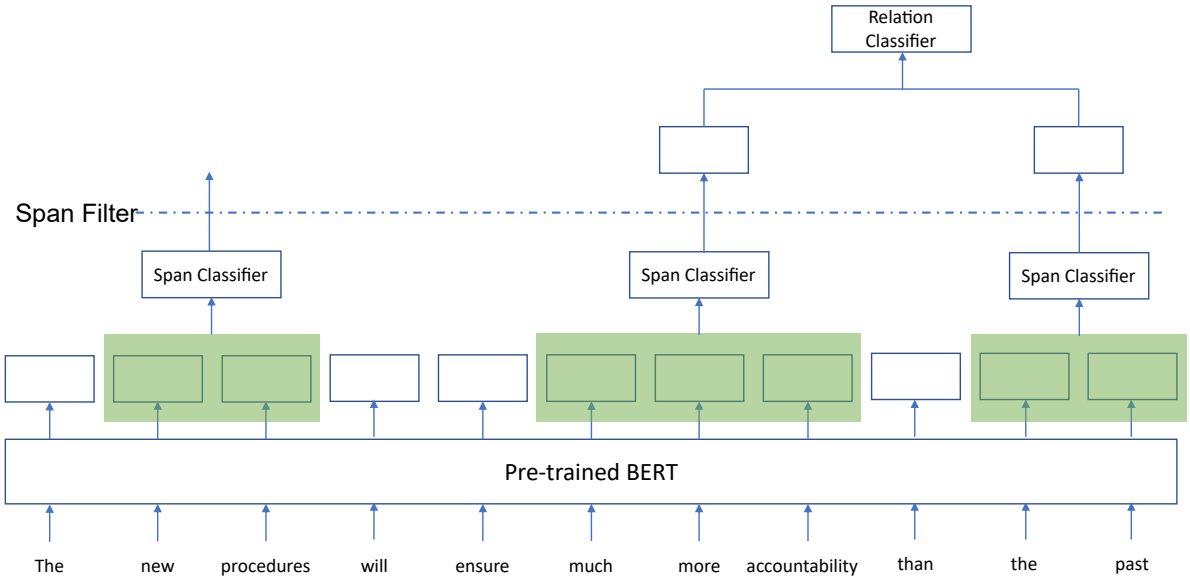

Figure 2.3: Causal SpERT model

2.4 Data Sets

We use 4 causality extraction data sets in our experiments. These data sets are converted into a common format (CREST from [8]) to ensure the same labeling and evaluation method. There is one cause and one effect span in each example for all data sets. To train the models, these data sets are preprocessed to different annotation templates from this common format.

Table 2.1: Data set statistics

Data Set	Train Size	Dev Size	Test Size
SCITE	4450	0	786
MedCaus	8881	2963	2959
BeCauSE	1226	0	354
FinCausal	1044	347	348

SemEval-2010 (SCITE) The SemEval-2010 data set was annotated by [16] to mark the noun phrases instead of single words. They added tags for multiple cause-effect pairs in a sentence. A new tag - Embedded Causality - was introduced to denote a cause or effect

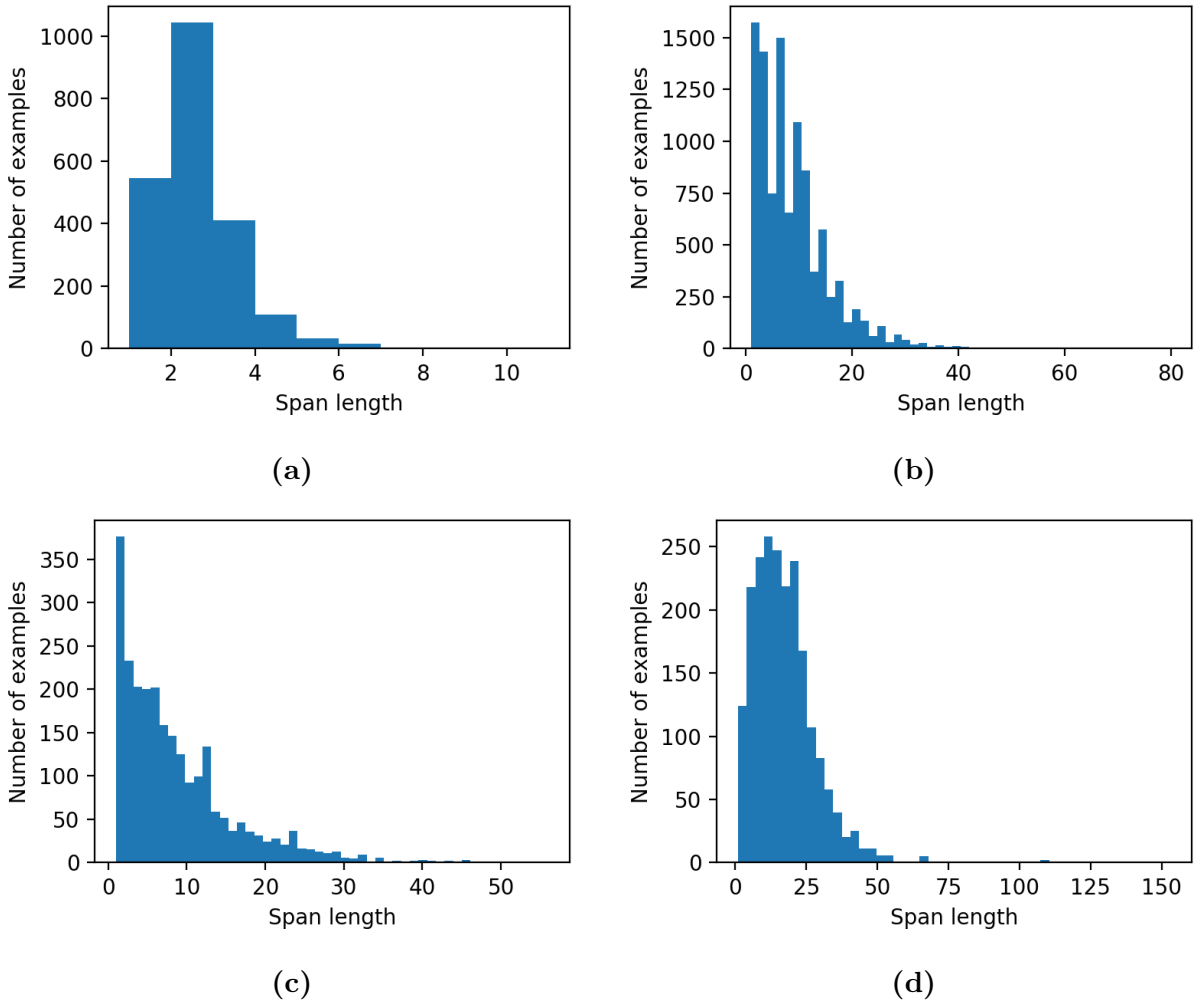

Figure 2.4: Distribution of span length for (a) SCITE, (b) MedCaus, (c) BeCauSE and (d) FinCausal

entity that is shared between multiple pairs. The train and test sets have 4450 and 786 examples respectively.

MedCaus This is a medical causality data set with 15000 sentences used in [6]. The examples were randomly extracted from medical articles in Wikipedia using manually selected causal cue words. The cause and effect entities are long phrases in this data set. There are 12064 causal examples and 3936 non-causal examples. The actual train, dev and test split used in that paper were not published. We randomly selected a 60-20-20 split of the whole data set after filtering examples with incorrect span labels.

BeCauSE The sentences for this data set [21] were collected from multiples sources: New York Times, Wall Street Journal, Penn Tree Bank and Congress hearings. The entities are also long phrases in this data set. There are 2150 examples in this data set. We filter duplicate examples containing the same sentence and instances with wrong span labels. After the preprocessing steps, we have 1570 total samples.

FinCausal This is a financial domain data set [22] [2] with 1750 examples. After discarding incorrect labels there are 1739 samples. The cause and effect spans in this data set are whole sentences. So the contexts in this data set are paragraphs with multiple sentences. The train, dev and test sets are selected randomly in a 60-20-20 split.

2.5 Experiments

Here, we discuss the performance of different models on the cause-effect extraction task on these 4 data sets.

2.5.1 Evaluation

The models are evaluated using micro F1 score for causality extraction. For sequence tagging models, the predicted tags are converted to spans by detecting the start and end from the tags. For IOBES tags, B and E denote the start and end. With BIO tags, the end can be detected by finding the end of B or I tags.

$$\text{Precision} = \frac{\text{no. of correct predicted spans}}{\text{total predicted spans}} \tag{2.7}$$

$$\text{Recall} = \frac{\text{no. of correct predicted spans}}{\text{total ground truth spans}} \tag{2.8}$$

$$\text{F1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{2.9}$$

Exact vs. Partial Match As there are multiple words in a cause or effect span, it is possible that the predicted span will match with some of the words in the ground truth span. We can stipulate that the predicted span must exactly match the ground truth span, i.e. the start

[2] https://github.com/yseop/YseopLab/tree/develop/FNP_2020_FinCausal/data

and end of the spans must be same. We term this method as "Exact Match". In "Partial Match", we count the number of common words between the predicted span and the ground truth span (Fig. 2.5). So, the model is not penalized for missing one or two words from the ground truth span.

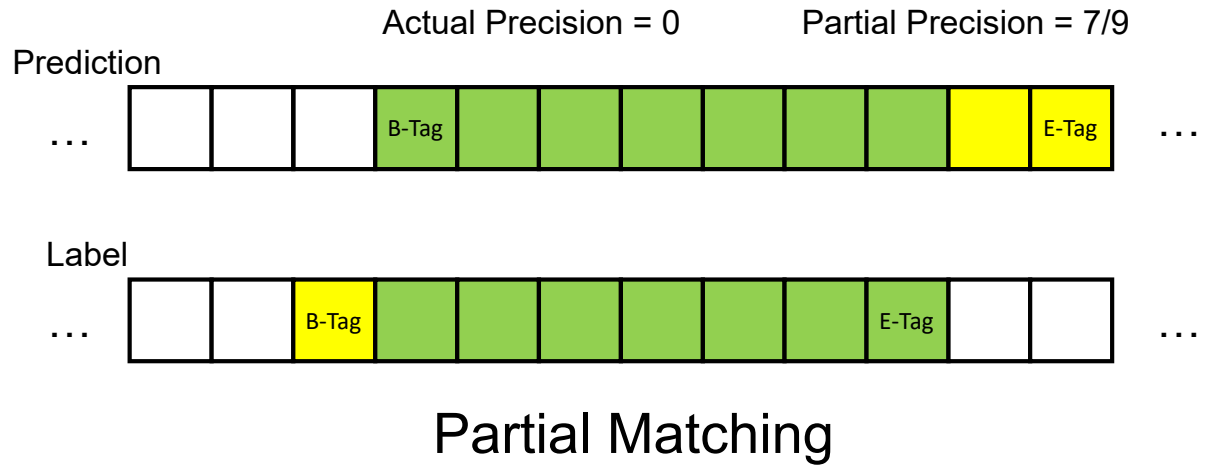

Figure 2.5: Exact versus partial match

All models have a better F1 score while using partial matching. The major difference in the predictions from sequence tagging and span based models is that the predicted spans cannot overlap in a sequence tagging model. With the span based model, we need to select a span from possibly overlapping predicted spans. In our experiments, we select the longest predicted span for each type.

2.5.2 Results

In the results tables, we highlight the best partial match scores in **bold** and the best exact match scores in *italic*.

SCITE We use the implementation of the model from [16][3] to run our experiments. The hyperparameters are set to the values reported in that paper. The batch size is 16 and the model is trained for 100 epochs for all data sets.

GCE The dependency parsing for this model is obtained using the `spacy` library. We train the GCE model with code from [6][4]. We set the batch size to 20 and trained the model on

[3]https://github.com/Das-Boot/scite
[4]https://github.com/farhadmfar/ace

causality extraction task for 50 epochs. All other hyperparameters are set to the values used in [6].

BERT The BERT sequence tagging model is implemented using the *transformers* [23][5] libray with PyTorch. We set the batch size to 16 and train the model for 40 epochs with cross entropy loss.

SpERT The SpERT model is trained with the implementation from [20][6]. We use the cause and effect spans as entities in the NER task. The relation classifier predicts if a causal relation exists in the input text. We train this model for 40 epochs with a batch size of 16. The number of negative entities is set to 10 and negative relations to 5 to create training samples. The model lists all possible spans of length less than the maximum span size and classifies them using the span classifier. For predicting the cause and effect span in a text segment, we select the longest predicted span for each type. The maximum span size, n is set to a value based on the distribution of span length in the data set (Fig 2.4). We found the 99 percentile span size from the training data distribution to work well across data sets (marked by * in the result tables).

Table 2.2: Precision, Recall and F1 score for causality extraction on SCITE data set (Partial matching score in bracket)

Model	Precision	Recall	F1
SCITE	88.19 (90.62)	89.65 (**89.77**)	88.91 (90.19)
GCE	87.64 (91.61)	83.51 (84.73)	85.52 (88.04)
BERT	90.50 (93.14)	*89.79* (89.51)	90.14 (**91.29**)
SpERT (n=4)	91.89 (**93.81**)	89.01 (88.09)	90.43 (90.86)
SpERT (n=6)*	*92.35* (92.52)	88.48 (86.16)	*90.37* (89.22)
SpERT (n=8)	91.87 (92.36)	88.74 (88.96)	90.28 (90.63)

2.6 Discussion

Here we analyse the relation of the performance of our trained models to several factors.

[5]https://github.com/huggingface/transformers
[6]https://github.com/lavis-nlp/spert

Table 2.3: Precision, Recall and F1 score for causality extraction on MedCaus data set (Partial matching score in bracket)

Model	Precision	Recall	F1
SCITE	59.00 (71.23)	54.74 (66.87)	56.79 (68.98)
GCE	54.31 (70.06)	56.97 (69.31)	55.61 (69.68)
BERT	61.12 (75.73)	60.38 (72.92)	60.75 (75.29)
SpERT (n=20)	*72.54* (**88.80**)	57.96 (65.09)	64.44 (75.12)
SpERT (n=25)	71.49 (86.54)	59.92 (72.05)	65.20 (78.63)
SpERT (n=30)	69.99 (85.90)	60.21 (75.75)	64.73 (80.51)
SpERT (n=32)*	71.52 (85.94)	*61.16* (**76.12**)	*65.93* (**80.73**)

Table 2.4: Precision, Recall and F1 score for causality extraction on BeCauSE data set (Partial matching score in bracket)

Model	Precision	Recall	F1
SCITE	37.74 (57.31)	36.78 (54.08)	37.25 (55.65)
GCE	27.15 (50.02)	21.89 (53.23)	24.24 (51.58)
BERT	51.46 (66.73)	52.40 (**68.29**)	51.92 (**67.51**)
SpERT (n=20)	53.92 (69.28)	46.61 (50.96)	50.00 (58.72)
SpERT (n=25)	54.01 (**70.49**)	50.42 (61.63)	52.15 (65.76)
SpERT (n=30)	54.68 (67.72)	51.98 (64.53)	53.29 (66.09)
SpERT (n=34)*	*55.13* (67.60)	*52.40* (66.26)	*53.73* (66.92)

2.6.1 Data Set Variation

Length of Spans The lengths of cause and effect spans in the SCITE data set are mostly below 5 with a very few spans longer than that (Fig. 2.4a). The difference in the performance of the BERT and SpERT models is very low - 0.29 for exact matching and 0.53 for partial matching - in this data set. On the other data sets with longer span lengths, we see that the SpERT model outperforms all sequence tagging models in terms of exact match. BERT has higher partial F1 score than BERT only on the BeCauSE data set. As partial matching score depends on the length of the span, longer spans can skew this score even though SpERT performs better in terms of exact matching. As the SpERT model predicts a contiguous span of words, it does not misclassify a span by missing 1 or 2 words in a long span. So the span based model has a clear advantage in data sets with longer span lengths.

Presence of Causal Connective Word We can check the data sets for the presence of common causal connective words - because, due to, lead to - for estimating the percentage of explicitly causal sentences. The MedCaus data set has the highest percentage of explicitly causal

Table 2.5: Precision, Recall and F1 score for causality extraction on FinCausal data set (Partial matching score in bracket)

Model	Precision	Recall	F1
SCITE	67.69 (79.41)	68.59 (80.13)	68.14 (79.77)
GCE	52.26 (69.04)	51.14 (75.52)	51.70 (72.14)
BERT	69.31 (77.24)	*71.08* (81.70)	70.18 (79.41)
SpERT (n=50)	69.70 (80.96)	66.71 (81.18)	68.18 (81.07)
SpERT (n=60)	*71.52* (**82.39**)	67.43 (83.19)	69.41 (**82.79**)
SpERT (n=64)*	70.61 (80.81)	66.57 (**83.81**)	68.53 (82.18)

sentences with common causal indicators - 51%. FinCausal has the lowest percentage at 8% but the causes and effects in this data set are also sentences instead of phrases or clauses. BeCauSE and SCITE data sets have similar percentage of explicit causality at 15% each. We see that BERT and SpERT models get close score on both these data sets while there is a large gap in the MedCaus data set. The GCE model also performs the best in the MedCaus data set. So, the presence of causality indicators help the SpERT and GCE models.

Average Word Frequency of Spans We use Wikipedia word frequencies to get the average word frequency of a cause or effect span. The FinCausal data set has the lowest word frequency as this is from the financial domain. MedCaus and BeCauSE data sets are similar in this metric and SCITE data set has the highest frequency. So the spans in SCITE data set have the most common words. We see that there is very little difference in the performance of the different models on this data set. In the FinCausal data set, the SCITE model does much better than GCE as it uses different character-based representations of words. It is clear that it is better to use data sets from different domains to evaluate the performance of these causality extraction models.

2.6.2 Model

The SpERT model has the best F1 score in terms of exact matching across all data sets. The advantage of this model is that we can select the maximum span size based on the distribution of the span lengths in the data set.

Parsing Information GCE model uses dependency parsing information to build the graph adjacency matrix used in the graph convolution layer (GCN) of this model. But this model does not perform as well as the BiLSTM-CRF (SCITE) model which only takes the word

sequence as input. So the use of word dependency relation does not play a significant role in these experiments.

Pre-trained Language Model It is evident from the performance of BERT based models 4 different data sets that pre-trained language models are much better than models only trained on the specific data set. A simple SoftMax classification layer on top of pre-trained BERT embeddings show significant improvement in F1 scores compared to both bi-directional LSTM and graph convolution encoder. As the data sets with labeled causes and effects have small number of training samples, it is important to augment the language comprehension ability of neural models using transfer learning methods.

CHAPTER 3
DOMAIN ADAPTATION USING LINGUISTIC INFORMATION FOR CAUSALITY EXTRACTION

3.1 Introduction

Extraction of cause and effect pairs from a sentence is an important step to recognize causal relationships and event sequences in a text corpus. It is not difficult to come up with rule based approaches for extracting causal information from sentences containing explicit causality indicators like *due to*, *as a result of* etc. But in the absence of specific patterns, we need to understand the meaning of the sentence in order to discover the spans of words that can be interpreted as causes and effects. Hence, we train neural models on the causality extraction task and adapt them to new domains to develop the ability to recognize these implicit patterns indicating causal relationships.

In this work, we address this information extraction problem with sequence-tagging neural network models. We use IOBES style tags to predict the cause and effect spans in a sentence. We focus on the challenge of adapting these models to unlabeled domains or data sets as the number of labeled data sets with causality labels is scarce. We hypothesize that understanding the linguistic pattern of causal sentences will help a model detect the causes and effects in a sentence even from a completely new domain. We adopt the adversarial domain adaptation framework (DANN) [7] to improve the performance of neural models on the cause-effect extraction task.

Sentence:	Canadian	wildfire	smoke	resulted	in	poor	air	quality	and
Label:	B-C	I-C	E-C	O	O	B-E	I-E	I-E	I-E
	haze	in	Northeast	US.					
	I-E	I-E	I-E	E-E					

Figure 3.1: Example sentence with IOBES style cause and effect tags

The unsupervised domain adaptation task has been heavily studied in image classification and segmentation problems but there is not much work on the causal information extraction problem. We propose a simple method for injecting linguistic information in the adversarial domain adaptation framework. Our experiments with linguitic information from

dependency parse shows improvements in the target domain performance compared to the baseline DANN method. We also employ an entropy measure in the domain classification loss function to encourage a direct relation of the domain classifier with the task classifier output. This results in significant improvement in target domain performance over multiple models and data sets.

3.2 Related Work

3.2.1 Causality Extraction

Majority of previous papers on causal knowledge extraction addresses the task of causal relation identification given a pair of entities. Traditional neural models are fused with external knowledge in [13] and [5]. A convolutional neural network is augmented with a word filter bank in [13] using words from causal frames in FrameNet [24][7]. These features are combined with the output of convolutional layers over word embeddings to predict causal relation between two entities in the sentence. External knowledge was used in [5] to enhance language models for event causality identification. A knowledge-aware model replaces event mentions with phrases containing knowledge from ConceptNet [4]. This model is combined with a mention-masking model to classify entity relations. Also, [11] injects external knowledge to a convolutional model by extracting different patterns from a 4-billion web page archive based on the cause and effect phrases.

A BiLSTM-CRF model was proposed in [16] to extract causality from text. They add a multi-head attention mechanism on top of the Bidirectional LSTM network for modeling the dependency of cause and effect words. In their experiments, pretrained Flair embeddings outperformed BERT and ELMO embeddings when concatenated with word embeddings as input to the BiLSTM-CRF model. A linguistically informed RNN architecture was trained in [14] for extracting causal relations from text. Different features like Parts of Speech, Dependency relations and WordNet attributes are used to create the linguistic feature input for the bidirectional LSTM model. They show improvement in cause and effect extraction task with the addition of linguistic information.

[7] https://framenet.icsi.berkeley.edu/

3.2.2 Domain Adaptation for Text

Most works on unsupervised domain adaptation for text data sets have addressed the sentiment analysis or text classification problem [25], [26], [27]. Adversarial domain adaptation for event recognition was studied in [28] but no improvement was shown beyond the DANN [7] method.

The only relevant work we found on domain adaptation of causlity extraction models is in [6]. They proposed a graph convolution network (GCN) that takes the dependency parse of a sentence as input. The output of GCN is fed to a BiLSTM layer to produce the final encoding for both causality identification and extraction. They use the DANN [7] adversarial domain adaptation method to improved the model performance in new domains.

3.3 Models

3.3.1 Adversarial Domain Adaptation

We adopt the DANN [7] approach for the baseline domain adaptation model. The main idea behind domain adaptation is to reduce the distributional shift between the source and target domain features. As we use the same task classifier for source and target domain examples, the features from the two domains should have a similar distribution for the task classifier to perform well in both domains.

Our sequence tagging model can be divided into two parts: the encoder f_θ and the token classifier h_ρ. The encoder converts the input sentence $X = [x_1, x_2, \ldots x_n]$ into a sequence of embeddings $[f_\theta(x_1), \ldots f_\theta(x_n)]$. The task classifier is the token classifier for this task. The goal of domain adaptation is to make the distribution of these encoder features similar across source and target domain. As we train the encoder f and token classifier h with labels from the source domain, the parameters of the model become tuned to the source specific patterns in the source domain. We want the model to learn domain invariant features in order to perform well on the target domain.

In DANN, a domain classifier is used to force the features learned by the encoder f_θ to be less specific to the source domain. The domain classifier g_ϕ predicts the domain of an example from the source or the target domain. Since this method was applied to object clasification, the same feature was used in both the task classifier f_θ and domain classifier g_ρ. In sequence tagging, we have a sequence of features for the token classifier but the domain classifier takes a representation of the sentence as input. This can be the encoder feature for

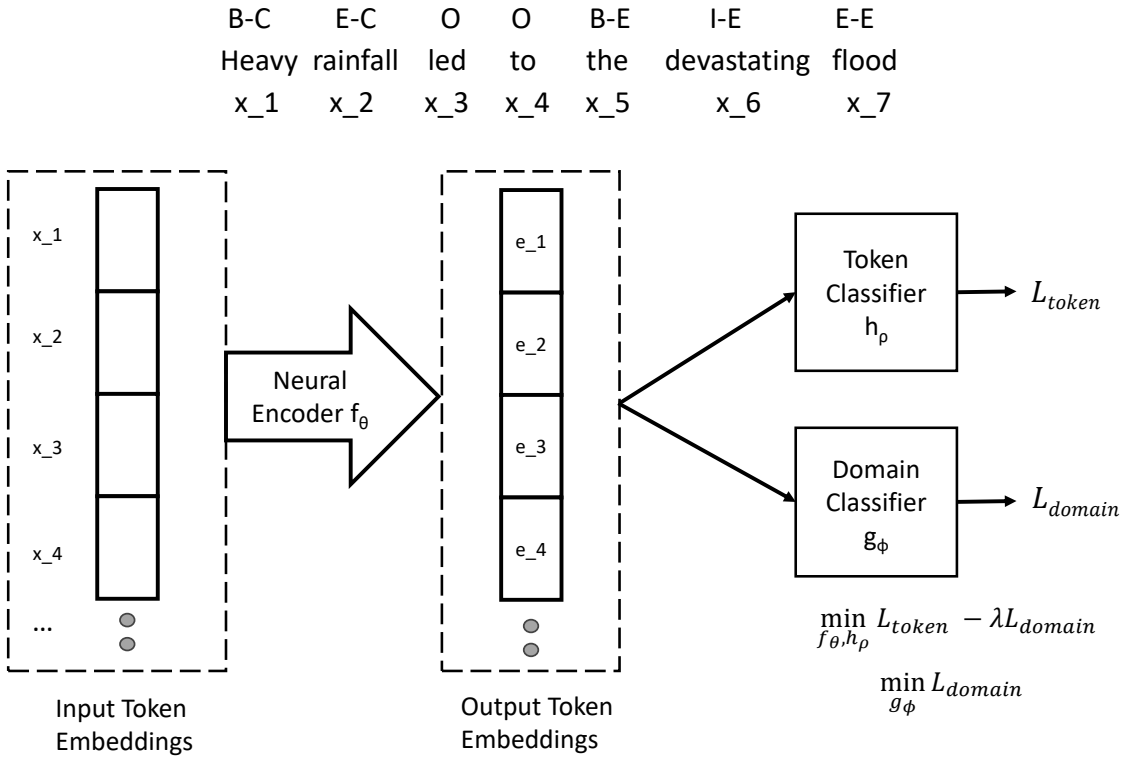

Figure 3.2: Adversarial domain adaptation for causality extraction for an example sentence with IOBES style cause, effect tags

the last token $f_\theta(x_n)$ in LSTM or the first token $f_\theta(x_1)$ in BERT. We use $f_\theta(X)$ to indicate the feature for the sentence.

There are two loss functions: the token classifier loss and the domain classifier loss.

$$\mathcal{L}_{token} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h_\rho(f_\theta(x_i)), y_i), X \in S \tag{3.1}$$

$$\mathcal{L}_{domain} = \mathcal{L}(g_\phi(f_\theta(X)), d_X), X \in S, T \tag{3.2}$$

where S, T are the source and target domains and d_X is the domain of the example X

The total loss is

$$\mathcal{L} = \mathcal{L}_{token} - \lambda \mathcal{L}_{domain} \tag{3.3}$$

For target domain examples, there is no label for the tokens. So, the token classification loss $\mathcal{L}_{token}$ is zero and we only have the domain classification loss $\mathcal{L}_{domain}$.

$$\hat{\theta}, \hat{\rho} = \arg\min_{\theta,\rho} \mathcal{L}_{token} - \lambda \mathcal{L}_{domain} \tag{3.4}$$

$$\hat{\phi} = \arg\min_{\phi} \mathcal{L}_{domain} \tag{3.5}$$

This minimization and maximization of loss is done simultaneously in [7] by taking gradient of the domain classification loss with respect to the encoder parameters and multiplying it by $-\lambda$. This is called the gradient reversal method.

3.3.2 Entropy Adaptation

We used the embedding of the sentence $f_\theta(X)$ as input to the domain classifier but it is not directly related to the token classification task. Although the first or last token embedding from the encoder f is related to other token embeddings due to the structure of the model, making this embedding invariant of the domain does not guarantee similar domain invariance in the features for other tokens in the sentence. Minimizing the entropy of the task classifier in the source and target domain iteratively has been effective for domain adaptation of image segmentation models in [29].

We can make the entropy of the token classifier domain invariant by using the adversarial training method with domain classifier. Instead of making the embedding of the sentence $f_\theta(X)$ invariant of the domain, we force the classification probability to have similar distribution in the source and target domain. The entropy for token classifier probability

$$p_i = softmax(h_\rho(f_\theta(x_i)) \tag{3.6}$$

$$H(p_i) = p_i \log(p_i) \tag{3.7}$$

$$\text{Entropy} = -\sum H(p_i) \tag{3.8}$$

where p_i is the token classifier output probability for token x_i

We use the vector $p\log(p)$ as the measure for entropy in our experiments. The input to the domain classifier $f_\theta(X)$ is replaced with entropy $H(p(x_i))$

$$\mathcal{L}_{domain} = \frac{1}{n} \sum_i \mathcal{L}(g_\phi(H(p_i)), d_{x_i}) \tag{3.9}$$

3.3.3 Linguistic Information

Dependency Parsing Relation We use the dependency parse relationship of the words in a sentence to label pairs of words in the sentence as related. In this parse, every word is related to one or more words in the sentence. We build a matrix of word pair relation in the sentence and randomly sample word pairs from unrelated pairs as negative samples. This is transformed into a binary classification loss for any word pair from the sentence. We represent the word pair using the output layer embedding and the sentence embedding.

$$f_\theta(x_i, x_j) = concat(f_\theta(x_i), f_\theta(x_j), f_\theta(X)) \tag{3.10}$$

$$\mathcal{L}_{parse} = \mathcal{L}(r_\pi(f_\theta(x_i, x_j)), y_{ij}) \tag{3.11}$$

where y_{ij} is the label for the relation between word i and j. We minimize this loss with respect to the model parameters f_θ and the relation classifier r_π.

POS Tag For the LSTM model we can use the POS tag of a word as an embedding. An embedding matrix is formed using the set of POS tags as a vocabulary. So, each POS tag has an embedding that is added to the word embedding to form the input embedding for the LSTM model.

$$e(x_i) = e_{pos}(x_i) + e_w(x_i) \tag{3.12}$$

where, e_w is the word embedding and e_{pos} is the POS tag embedding.

3.3.4 Modeling Contextual Information

In the adversarial domain adaptation method, the domain classifier guides the feature extractor to generate features that are indistinguishable between source and target domain examples. As our task is extracting cause and effect, the performance of the token classifier in the target domain is the key metric. We hyposthesize that incorporating a represenatation of the contextual information from the feature extractor into the token classifier will improve its performance in the target domain.

To model the contextual information, we add the embedding of the sentence or context

as input to the token classifier. We replace the classification prediction

$$h_\rho(f_\theta(e_t)) \tag{3.13}$$

with conditioning on the sentence embedding

$$h_\rho(f_\theta(e_t)|e_s) \tag{3.14}$$

f_θ is the feature extraction and h_ρ is the token classifier e_t, output embedding of token t e_s, output embedding of sentence s

This conditional classification model is implemented by concatenating the context embedding with the token embedding

$$h_\rho(f_\theta(e_t||e_s)) \tag{3.15}$$

The context embedding e_s is the last token embedding for LSTM and CLS token embedding for BERT.

3.4 Experiments

We evaluate the ability of a neural model to adapt to a unlabeled target data set while being trained with the location of causes and effects in the source data set. Hence we require data sets with cause-effect labels for our experiments. We convert the data sets to the CREST [8] format to simplify the causality location and label format for all data sets. We use the MedCaus data set as source, the BeCauSE and SemEval data sets as target domains for our experiments.

We use the parser from the *spacy* [30] library trained on the *en_core_web_sm* corpus to generate dependency parse for sentences from both source and target data sets. We also get parts-of-speech (POS) tags for every word in a sentence from the spacy parser. We build the word embedding matrix for LSTM with 300 dimensional pre-trained glove embeddings for words from the training set. So, the test set contains more out-of-vocabulary (OOV) words than the train set.

The bidirectional LSTM model has 1 layer with hidden size of 100 dimensions. We

add a dropout layer with probability of 0.5 after word embedding layer, LSTM layers and output classifiers. The token classifier, parsing relation classifier and domain classifiers are all 2-layer MLP with an intermediate layer having the same dimension as the LSTM hidden size. The input and output layer dimension for the 2-layer MLP of these classifiers depend on their respective inputs and outputs. For BERT-base models, we use 2-layer MLP with intermediate layer size same as the transformer hidden size.

We use IOBES style tags for the sequence tagging models following [6]. So the token classifier predicts probabilities for 9 classes. The relation classifier and domain classifier are both binary classifiers with predictions over 2 classes. We use the validation set F1 score as the measure for early stopping. We tune the learning rate and the weights for the domain classification loss and parsing relation classification loss based on the validation set performance.

3.4.1 Data Sets

MedCaus The MedCaus data set was introduced in [6] with 15000 sentences labeled with the location of causes and effects. These sentences were randomly extracted from medical articles in Wikipedia. The data set contains both causal and non-causal sentences with approximately 26% non-causal examples. We preprocessed and converted this data set into the CREST format. We removed sentences with different labeling issues like overlap between cause and effect spans, incorrect missing token index etc. This data set was not published with a train, dev, test split. We used a 60-20-20 split for the data set resulting in 8882 training, 2943 validation and 2946 test samples.

BeCauSE The sentences for this data set [21] were collected from multiples sources: New York Times, Wall Street Journal, Penn Tree Bank and Congress hearings. The entities are also long phrases in this data set. There are 2150 examples in this data set. We filter duplicate examples containing the same sentence and instances with wrong span labels. After the preprocessing steps, we have 1613 total samples divided into 1130 training, 125 dev and 358 test samples.

SemEval The SemEval-2010 data set was introduced by [15] to be used in the shared tasks. They labeled different types of relationships between word pairs in a sentence including the cause-effect relationship. For our experiments, we use the sentences with cause-effect

Table 3.1: Data set statistics

Data Set	Train	Dev	Test	Span Length Mean (Variance)
MedCaus	8882	2943	2946	9.01 (50.13)
BeCauSE	1130	125	358	8.74 (51.64)
SemEval	1600	200	200	1.05 (0.05)
CausalNewsCorpus	891	99	110	8.27 (24.09)

relations as positive examples and other relations as negative examples. Similar to [6], we select 1000 positive examples and 1000 negative examples by random sampling. After a 80-10-10 split, we have 1600 train, 200 dev and 200 test samples.

CausalNewsCorpus Data Set This data set contains causal relations in event sentences from News sources. The sentences are annotated with cause, effect and signal labels. The *signal* label denotes the words or phrases which indicate a causal relationship in the sentence. We only use the cause and effect labels for our experiments to compare with other data sets. New version (version 2) [31][8] of this data set contains 990 training examples, 110 dev examples. We use the dev set as test set for our experiments.

3.5 Results on Domain Adaptation

We compare the performance of different models for the domain adaptation task here. Although, the GCE [6] model reported domain adaptation results with the DANN method for the cause-effect extraction task, we did not get similar results with their model. We used their code for the Graph Convolutional Network model but we had to implement the domain adaptation method as their code for domain adaptation in the github repo [6][9] was not complete. We use the DANN method as baseline for our experiments denoted with the *-Adv* suffix. For a model X, we use the following naming scheme for different methods:

Table 3.2: Model names and corresponding loss functions

X-Adv	Loss $\mathcal{L}_{adv} = \mathcal{L}_{token} - \lambda\mathcal{L}_{domain}$	Sec 3.3.1
X-Parse-Adv	Loss $\mathcal{L}_{parse-adv} = \mathcal{L}_{adv} + \mathcal{L}_{parse}$	Sec 3.3.3
X-Entropy-Adv	Loss $\mathcal{L}_{entropy} = \mathcal{L}_{token} - \lambda\mathcal{L}_{domain-entropy}$	Sec 3.3.2
X-Parse-Entropy-Adv	Loss $\mathcal{L}_{parse-entropy} = \mathcal{L}_{entropy} + \mathcal{L}_{parse}$	Sec 3.3.2 and Sec 3.3.3
X-Ctx-Adv	Loss $\mathcal{L}_{ctx-adv} = \mathcal{L}_{token-ctx} - \lambda\mathcal{L}_{domain}$	Sec 3.3.1 and Sec 3.3.4

[8]https://github.com/tanfiona/CausalNewsCorpus/tree/master/data/V2
[9]https://github.com/farhadmfar/ace/

We also use the simple baseline of training a model on the source domain only without domain adaptation listed as X (Source) and the same model trained on the target domain labels as X (Target). These baselines should form the lower and upper bound for the unsupervised domain adaptation task. In the results tables, we highlight the best partial match scores in **bold** and the best exact match scores in *italic*.

3.5.1 Evaluation of Transformer Models

Here, we explain the evaluation process for matching the predicted cause and effect spans with the ground truth. For transformers, the label for a token is converted to word level by selecting the predicted label of the first token in a word. To calculate the precision and recall, we extract all possible spans predicted by the model and compare them with the ground truth. If the model predicts multiple spans, all of them are compared with labeled spans. So, the number of spans predicted by the model have an effect on the precision, recall score.

Span Matching For a sentence with labeled cause and effect spans $\{s_g\}$, we find the intersection with all predicted spans $\{s_p\}$ of the same type

$$\text{match}_{\text{score}} = \sum_i n(s_g^i \cap s_p^i) \tag{3.16}$$

where i denotes the type of span from {cause, effect}
The total number of ground truth spans is

$$\text{total}_{\text{label}} = \sum_i n(s_g^i) \tag{3.17}$$

The total number of predicted spans is

$$\text{total}_{\text{pred}} = \sum_i n(s_p^i) \tag{3.18}$$

Here, we defince the precision, recall and F1 scores. Precision score for all examples

$$Prec = \frac{\sum \text{match}_{\text{score}}}{\sum \text{total}_{\text{pred}}} \tag{3.19}$$

Recall score for all examples

$$Rec = \frac{\sum \text{match}_{\text{score}}}{\sum \text{total}_{\text{label}}} \tag{3.20}$$

F1 score for all examples

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec} \tag{3.21}$$

3.5.2 Bidirectional LSTM Model

We report the performance of a bidirectional LSTM model trained on source domain data only in the first row. The last row shows the biLSTM model trained on the target data set. We use the last token embedding from LSTM as sentence embedding for training the domain classifier. Adversarial domain adaptation improves the partial recall score of the model on the target data set. We see an increase in partial F1 score from 7.13 to 37.16 after applying the DANN adaptation method to BiLSTM. The use of entropy of the token classifier in domain classification also brings improvement in the target domain performance. With the BeCauSE data set as target, the combination of entropy loss and parsing loss with POS tag embedding gives the best partial F1 score of 42.29 compared to the baseline domain adaptation model score 37.16. In the SemEval data set, the combination of parsing and entropy loss increases the F1 score from 20.4 to 22.8. In the CausalNewsCorpus data set, the parsing loss (BiLSTM-Parse-Adv) gives the best partial F1 score while the BiLSTM-Parse-Entropy-Adv model has the best exact F1 score and competitive partial F1 score.

Table 3.3: MedCaus to BeCauSE adaptation with BiLSTM

Model	Precision	Recall	F1
BiLSTM (Source)	0 (22.54)	0 (4.24)	0 (7.13)
BiLSTM-Adv	10.72 (43.88)	7.68 (32.23)	8.95 (37.16)
BiLSTM-Parse-Adv	8.63 (46.26)	6.70 (31.88)	7.55 (37.75)
BiLSTM-Entropy-Adv	12.37 (44.39)	8.24 (33.15)	9.89 (37.95)
BiLSTM-Ctx-Entropy-Adv	12.45 (45.01)	9.22 (32.08)	10.59 (37.46)
BiLSTM-Parse-Entropy-Adv	12.57 (49.89)	8.52 (26.83)	10.16 (34.89)
BiLSTM-Parse-POS-Entropy-Adv	*17.98* (**55.87**)	*11.17* (**34.03**)	*13.78* (**42.29**)
BiLSTM (Target)	15.36 (48.00)	16.06 (41.33)	15.70 (44.42)

Table 3.4: MedCaus to SemEval adaptation with BiLSTM

Model	Precision	Recall	F1
BiLSTM (Source)	0 (2.45)	0 (2.40)	0 (2.43)
BiLSTM-Adv	7.24 (13.39)	5.50 (43.27)	6.25 (20.45)
BiLSTM-Parse-Adv	3.16 (11.83)	3.00 (58.65)	3.08 (19.69)
BiLSTM-Entropy-Adv	*10.89* (**17.47**)	*5.5* (24.5)	*7.3* (20.4)
BiLSTM-Parse-Entropy-Adv	3.97 (14.15)	3.50 (58.65)	3.72 (**22.8**)
BiLSTM-Parse-Ctx-Entropy-Adv	2.69 (11.42)	3.00 (**60.58**)	2.84 (19.22)
BiLSTM-Parse-POS-Entropy-Adv	5.97 (16.39)	4.00 (28.85)	4.79 (20.91)
BiLSTM (Target)	74.14 (77.84)	64.50 (65.87)	68.98 (71.35)

Table 3.5: MedCaus to CausalNewsCorpus adaptation with BiLSTM

Model	Precision	Recall	F1
BiLSTM (Source)	0 (35.21)	0 (2.68)	0 (4.98)
BiLSTM-Adv	12.31 (52.19)	7.27 (19.63)	9.14 (28.53)
BiLSTM-Parse-Adv	6.25 (51.18)	5.45 (33.49)	5.82 (40.49)
BiLSTM-Parse-Ctx-Adv	*13.45* (**63.39**)	*10.45* (**34.62**)	*11.76* (**44.78**)
BiLSTM-Entropy-Adv	9.84 (45.22)	5.45 (22.45)	7.02 (30.01)
BiLSTM-Parse-Entropy-Adv	8.89 (53.48)	7.27 (31.52)	8.00 (39.66)
BiLSTM-Parse-POS-Entropy-Adv	5.97 (16.39)	4.00 (28.85)	4.79 (20.91)
BiLSTM (Target)	25.09 (65.02)	29.09 (66.13)	26.95 (65.57)

3.5.3 BERT Model

For domain adaptation with BERT, we use the CLS token embedding as the sentence representation to be adapted with the domain classifier. The BERT model performs very well on the target data set without domain adaptation compared to BiLSTM. The supervised model performance is also much higher with BERT noted in the last row as BERT (Target). We see improvement in target domain F1 score when the entropy loss is used with the BERT-base model. With BeCauSE data set as target, we get a 44.59 F1 score compared to the 37.18 baseline score. For the SemEval target data set, the F1 score increases to 26.79 from 25.75. Using the CausalNewsCorpus data set as target, we get similar results. The BERT-Entropy-Adv model has the best partial F1 score. The entropy loss works better without the dependency parsing loss.

Table 3.6: MedCaus to BeCauSE adaptation with BERT

Model	Precision	Recall	F1
BERT (Source)	39.77 (66.42)	14.39 (23.71)	21.13 (34.95)
BERT-Adv	3.40 (29.31)	1.82 (50.82)	2.37 (37.18)
BERT-Parse-Adv	35.14 (71.72)	9.08 (15.27)	14.43 (25.17)
BERT-Entropy-Adv	*43.45* (**67.67**)	*20.39* (**33.24**)	*27.76* (**44.59**)
BERT-Ctx-Entropy-Adv	41.31 (67.40)	17.60 (28.81)	24.68 (40.36)
BERT-Parse-Entropy-Adv	43.05 (70.99)	18.16 (29.73)	25.54 (41.92)
BERT (Target)	42.46 (61.10)	48.74 (63.67)	45.84 (62.36)

Table 3.7: MedCaus to SemEval adaptation with BERT

Model	Precision	Recall	F1
BERT (Source)	0 (73.28)	0 (12.48)	0 (21.33)
BERT-Adv	*5.36* (15.97)	*4.50* (66.35)	*4.89* (25.75)
BERT-Parse-Adv	4.08 (**17.44**)	3.00 (57.69)	3.46 (**26.79**)
BERT-Entropy-Adv	6.13 (16.49)	5.00 (65.87)	5.51 (26.37)
BERT-Ctx-Entropy-Adv	5.84 (15.49)	4.50 (65.38)	5.08 (25.02)
BERT-Parse-Entropy-Adv	4.29 (15.63)	3.50 (**67.79**)	3.86 (25.41)
BERT (Target)	79.21 (82.13)	80.0 (81.73)	79.60 (81.93)

Table 3.8: MedCaus to CausalNewsCorpus adaptation with BERT

Model	Precision	Recall	F1
BERT (Source)	26.98 (67.62)	7.73 (15.22)	12.01 (24.85)
BERT-Adv	33.33 (**100**)	0.90 (1.98)	1.77 (3.89)
BERT-Parse-Adv	33.72 (77.76)	13.18 (21.54)	18.96 (33.74)
BERT-Entropy-Adv	*41.33* (75.72)	14.09 (**24.06**)	21.02 (**36.52**)
BERT-Ctx-Entropy-Adv	37.80 (68.24)	14.09 (23.26)	20.53 (34.69)
BERT-Parse-Entropy-Adv	39.76 (74.58)	*15.00* (23.47)	*21.78* (36.02)
BERT (Target)	49.45 (78.37)	61.36 (80.39)	54.77 (79.37)

3.5.4 GPT2 Model

We use the GPT2 model with pretrained weights [23][10] in the adversarial domain adaptation framework. Since GPT2 is an autoregressive model, there is no CLS embedding like BERT to be used as the context embedding. We use the last token embedding as the input to the domain classifier similar to the way we trained the LSTM model. GPT2 performs like BERT in the different domain adaptation configurations. The use of context embedding with entropy loss model, GPT2-Ctx-Entropy-Adv has better partial F1 score compared to the combination of entropy and parsing loss, GPT2-Parse-Entropy-Adv, on all target data

[10]https://huggingface.co/gpt2

sets.

Table 3.9: MedCause to BeCauSE adaptation with GPT2

Model	Precision	Recall	F1
GPT2 (Source)	5.81 (56.74)	5.31 (22.57)	5.55 (32.29)
GPT2-Adv	4.75 (47.81)	1.96 (8.39)	2.77 (14.27)
GPT2-Parse-Adv	7.73 (56.27)	4.61 (15.82)	5.77 (24.69)
GPT2-Entropy-Adv	5.06 (54.92)	4.75 (23.26)	4.89 (32.68)
GPT2-Ctx-Entropy-Adv	*12.84* (**63.37**)	*9.92* (**28.56**)	*11.19* (**39.37**)
GPT2-Parse-Entropy-Adv	6.04 (48.91)	5.45 (22.05)	5.73 (30.39)
GPT2 (Target)	11.66 (55.52)	20.67 (54.91)	14.91 (55.21)

Table 3.10: MedCaus to SemEval adaptation with GPT2

Model	Precision	Recall	F1
GPT2 (Source)	1.72 (51.38)	1.53 (10.19)	1.62 (17.01)
GPT2-Adv	2.34 (57.63)	2.04 (13.33)	2.18 (21.65)
GPT2-Parse-Adv	*7.61* (9.51)	*11.00* (**48.08**)	*8.99* (15.87)
GPT2-Entropy-Adv	1.94 (60.30)	1.53 (10.61)	1.71 (18.04)
GPT2-Ctx-Entropy-Adv	0.44 (**87.14**)	0.26 (16.37)	0.32 (**27.56**)
GPT2 (Target)	60.51 (63.96)	59.00 (60.58)	59.75 (62.22)

Table 3.11: MedCaus to CausalNewsCorpus adaptation with GPT2

Model	Precision	Recall	F1
GPT2 (Source)	3.43 (65.91)	3.18 (23.20)	3.30 (34.32)
GPT2-Adv	4.07 (60.18)	3.18 (21.22)	3.57 (31.38)
GPT2-Parse-Adv	4.09 (64.99)	2.27 (13.83)	2.92 (22.80)
GPT2-Entropy-Adv	4.48 (59.57)	2.73 (11.84)	3.39 (19.76)
GPT2-Ctx-Entropy-Adv	*18.11* (**69.27**)	*10.45* (**24.76**)	*13.26* (**36.48**)
GPT2-Parse-Entropy-Adv	7.09 (63.62)	5.00 (18.65)	5.86 (28.84)
GPT2 (Target)	15.33 (75.51)	32.27 (75.51)	20.79 (75.11)

3.5.5 Discussion

The performance of the models compared in our experiments show a significant variance over different dataset pairs. For the MedCaus to BeCauSE adaptation, all models have very good performance. Using the SemEval dataset as target, we see that different domain adaptation methods provide comparably smaller improvement over the baseline of no domain adapation. The distribution of the span lenghts in the data sets explain this behaviour.

While the MedCaus and BeCauSE data sets have similar span lengths, the SemEval data set contains 1 or 2 word spans resulting in a significant difference in the distribution of the IOBES tags. We get better performance on the CausalNewsCorpus data set, specially with the BiLSTM model where the span lengths are comparable to the source domain.

CHAPTER 4
CONCLUSION

In this work, we combined data sets from different domains with different styles of cause and effect labels into a standard format. We applied a span based approach to extracting causal knowledge that works better than the sequence tagging models across data sets from different domains. Span based models have the following advantages over sequence tagging models: 1) We can tune the maximum span threshold for each domain/dataset, which enables the span-based model to consistently achieve the best performance across the domains and 2) A span based model can handle multiple cause and effect entities. We examined the effect of modeling methods like pre-training, parsing based neural layer, and data set attributes like span size distributions on the model performance.

We also improved the unsupervised domain adaptation performance of adversarial domain adaptation methods for the task of causal information extraction. We showed that by requiring the task specific classifier output to have similar distributions across two domains we can improve the domain adaptation performance of both word based models like LSTM and pre-trained transformer models like BERT and GPT2. In adversarial domain adaptation, we should focus on the specific task instead of the distribution of the model representation. We also found siginificant improvement in domain adaptation performance when training the model to recogninze word dependency relations in both domains simultaneously. The similarity of sentences and the cause-effect spans in the source and target domain influence the effectiveness of our domain adaptation methods for all models. These results give us key insights in our quest to develop a general causal information extraction model.

REFERENCES

[1] C. Hidey and K. McKeown, "Identifying causal relations using parallel wikipedia articles," in *Proc. of the 54th Annu. Meet. of the Assoc. for Comput. Ling.*, vol. 1: Long Papers, 2016, pp. 1424–1433.

[2] L. Gao, P. K. Choubey, and R. Huang, "Modeling document-level causal structures for event causal relation identification," in *Proc. of the 2019 Conf. of the North Amer. Chap. of the Assoc. for Comput. Ling.: Human Lang. Technol.*, vol. 1: Long and Short Papers, 2019, pp. 1808–1817.

[3] X. Zuo, Y. Chen, K. Liu, and J. Zhao, "Towards causal explanation detection with pyramid salient-aware network," in *China Nat. Conf. on Chin. Comput. Ling.*, 2020, pp. 113–128.

[4] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: an open multilingual graph of general knowledge," in *Proc. of the 31st AAAI Conf. on Artif. Intell.*, 2017, pp. 4444–4451.

[5] J. Liu, Y. Chen, and J. Zhao, "Knowledge enhanced event causality identification with mention masking generalizations," in *Proc. of the 29th Int. Conf. on Int. Joint Conf. on Artif. Intell.*, 2021, pp. 3608–3614.

[6] F. Moghimifar, G. Haffari, and M. Baktashmotlagh, "Domain adaptative causality encoder," in *Proc. of the The 18th Annu. Workshop of the Australas. Lang. Technol. Assoc.*, 2020, pp. 1–10.

[7] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. of the 32nd Int. Conf. on Int. Conf. on Mach. Learn.*, vol. 37, 2015, pp. 1180–1189.

[8] P. Hosseini, D. A. Broniatowski, and M. Diab, "Predicting directionality in causal relations in text," 2021, *arXiv:2103.13606*.

[9] J. Xu, W. Zuo, S. Liang, and X. Zuo, "A review of dataset and labeling methods for causality extraction," in *Proc. of the 28th Int. Conf. on Comput. Ling.*, 2020, pp. 1519–1531.

[10] E. Blanco, N. Castell, and D. Moldovan, "Causal relation extraction," in *Proc. of the 6th Int. Conf. on Lang. Resour. and Eval., LREC*, 2008, pp. 310–313.

[11] 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," in *Proc. of the 31st AAAI Conf. on Artif. Intell.*, 2017, pp. 3466–3473.

[12] S. Zhao, T. Liu, S. Zhao, Y. Chen, and J.-Y. Nie, "Event causality extraction based on connectives analysis," *Neurocomputing*, vol. 173, no. P3, pp. 1943–1950, Jan 2016.

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

[14] T. Dasgupta, R. Saha, L. Dey, and A. Naskar, "Automatic extraction of causal relations from text using linguistically informed deep neural networks," in *Proc. of the 19th Annu. SIGdial Meet. on Discourse and Dialogue*, 2018, pp. 306–316.

[15] 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 *Proc. of the 5th Int. Workshop on Semantic Eval.*, 2010, pp. 33–38.

[16] Z. Li, Q. Li, X. Zou, and J. Ren, "Causality extraction based on self-attentive bilstm-crf with transferred embeddings," *Neurocomputing*, vol. 423, pp. 207–219, Jan 2021.

[17] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. of the 27th Int. Conf. on Comput. Ling.*, 2018, pp. 1638–1649.

[18] A. Komninos and S. Manandhar, "Dependency based embeddings for sentence classification tasks," in *Proc. of the 2016 Conf. of the North Amer. Chap. of the Assoc. for Comput. Ling.: Human Lang. Technol.*, 2016, pp. 1490–1500.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North Amer. Chap. of the Assoc. for Comput. Ling.: Human Lang. Technol., Vol. 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[20] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," in *Proc. of the 24th Eur. Conf. on Artif. Intell.*, 2020, pp. 2006–2013.

[21] J. Dunietz, L. Levin, and J. G. Carbonell, "The because corpus 2.0: Annotating causality and overlapping relations," in *Proc. of the 11th Ling. Annot. Workshop*, 2017, pp. 95–104.

[22] D. Mariko, H. Abi Akl, E. Labidurie, S. Durfort, H. De Mazancourt, and M. El-Haj, "The financial document causality detection shared task (fincausal 2020)," in *Proc. of the 1st Joint Workshop on Financ. Narrat. Process. and MultiLing Financ. Summar.*, 2020, pp. 23–32.

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. of the 2020 Conf. on Empirical Methods in Natural Lang. Process.: Syst. Demo.*, 2020, pp. 38–45.

[24] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proc. of the 36th Annu. Meet. of the Assoc. for Comput. Ling. and 17th Int. Conf. on Comput. Ling.*, vol. 1, 1998, pp. 86–90.

[25] N. N. Trung, L. N. Van, and T. H. Nguyen, "Unsupervised domain adaptation for text classification via meta self-paced learning," in *Proc. of the 29th Int. Conf. on Comput. Ling.*, 2022, pp. 4741–4752.

[26] D. Wright and I. Augenstein, "Transformer based multi-source domain adaptation," in *Proc. of the 2020 Conf. on Empirical Methods in Natural Lang. Process. (EMNLP)*, 2020, pp. 7963–7974.

[27] C. Du, H. Sun, J. Wang, Q. Qi, and J. Liao, "Adversarial and domain-aware bert for cross-domain sentiment analysis," in *Proc. of the 58th Annu. Meet. of the Assoc. for Comput. Ling.*, 2020, pp. 4019–4028.

[28] A. Naik and C. Rose, "Towards open domain event trigger identification using adversarial domain adaptation," in *Proc. of the 58th Annu. Meet. of the Assoc. for Comput. Ling.*, 2020, pp. 7618–7624.

[29] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. of the IEEE/CVF Conf. on Comput. Vision and Pattern Recognit.*, 2019, pp. 2517–2526.

[30] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," spaCy.io. Accessed: August 30, 2021. [Online]. Available: https://spacy.io/

[31] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, T. Caselli, O. Uca, F. F. Liza, and N. Oostdijk, "Event causality identification with causal news corpus-shared task 3, case 2022," in *Proc. of the 5th Workshop on Chall. and Appl. of Automat. Extract. of Socio-political Events from Text (CASE)*, 2022, pp. 195–208.