# Event Causality Detection using NLP Techniques

## Semester Project Report

Spring 2022

By: SHAN Gaganjot Kaur

**Supervisor(s):** Pasquale Lisena, Youssra Rebboud, Raphaël Troncy

27 June 2022

# 1. Introduction

Detecting events and their relationships has become a crucial part of natural language processing. It has endless applications in the information research field while still being a highly challenging task. An event can be defined as something that happens. It can have many questions associated with it, for instance, when, where, how, by whom, etc. The goal of event detection is to identify events and their relationships in unstructured data. The relationships between two events can be further classified into temporal and causal. Events are temporal when they are time-related. For instance, "We went to school today," or "he slept at 9 p.m. tonight." Here, "9 pm tonight" is included in "today" and hence "going to school" and "sleeping" are temporally related. Whereas, when one relationship is caused by the other, the type of the relationship is called a causal relationship. For example, "his brakes failed and he had an accident." Here, failure of the brakes caused the accident; thus the events are causally related.

Several techniques have been proposed so far for the detection of relationships that are either exclusively based on machine learning, linguistic markers, or others. In this project, we have studied a knowledge-based neural network, K-NN ([Pengfei Li and Kezhi Mao, 2019]()), for casual relationship detection and other relation extraction. The K-NN architecture consists of a knowledge-oriented channel in addition to the neural network, data-oriented channel. The knowledge channel is generated from lexical knowledge bases such as WordNet and FrameNet. On the other hand, the sieve-based model CATENA (Mirza and Tonelli, 2018), deals with both CAusal and TEmporal relation extraction. CATENA has a multiclass architecture for the extraction and classification of both temporal and causal events.

CATENA takes any annotated document with DCT, events, and timexs as an input and outputs the same document with temporal and causal links between the events in CATENA. DCT stands for Document Creation Time for the respective document, an event can be interpreted as any occurrence such as "eating", "establishing", etc., and timexs represent any temporal information, for example, "2 AM ", "Friday", "2022", etc. and outputs the same document with temporal and causal links between each event for CATENA.

Whereas, In Knowledge-oriented neural networks[1], we are working on the SemEval-2010 task 8 challenge for the classification of the entity pairs. An event is expressed as a sentence involving keywords and actions related to those keywords. The input sentences are marked with target entities e1 and e2. For instance, given a sentence, S: "The financial <e1>crisis</e1> resulted in 12% <e2>unemployment<e2/> in this country", the entity pair is defined as (e1,e2). The sentences in the database are the reflections of events, which include the relationship between the entities (e1 and e2) explicitly with linguistic clues; "results in", "cause", and "because".

Our end goal in this project is to perform relation extraction and classification using different approaches. We are mainly working on causal and temporal relations. In addition, we want to study the architecture(s) and fine-tune model's performance.

[1] https://github.com/gaganjotshan/Event_Detection

# 2. Knowledge-oriented Neural Network

Relationship extraction is a subfield of information extraction and is used to extract relationships from natural language texts. Extracting a relationship is a challenge that has several existing approaches, either rule-based (non-statistical) or machine-learning-based (statistical) methods.

Each of these methods has some issues to encounter. For instance, a rule-based approach requires a lot of manual work to construct patterns for the analysis of relationships; the statistical method, on the other hand, relies on a large amount of data that requires sophisticated feature engineering. The performance of the system strongly depends on the quality of the designed features. The NLP toolkits used for the process are imperfect and can cause error propagation to the relation extraction system. To tackle the feature extraction problem in a sophisticated manner in deep learning, models with pre-trained embeddings have been used.

To have a sophisticated approach to the feature extraction task ([2]Nguyen & Grishman, 2015; Santos, Xiang, & Zhou, 2015), they achieved state-of-the-art performance for the SemEval-2010 task 8 challenge by constructing deep learning models for relation extraction using convolutional neural networks (CNN) with pre-trained word embeddings. Pre-trained word embeddings encode semantic and syntactic information about words into fixed-length vectors. ([3]Mikolov, Chen, Corrado, & Dean, 2013; Turian, Ratinov, & Bengio, 2010). Despite being able to perform better feature extraction, these models largely depend on huge amounts of data and demand to encapsulate all the possible causal relation expressions in the natural language.

Moreover, these models with a large number of free parameters easily tend to overfit the training data, so the performance of deep learning models is compromised. Pengfei Li Kezhi Mao: K-CNN is an approach developed to tackle these issues by introducing a knowledge-oriented channel to the state-of-the-art deep learning models ([4]Pengfei Li Kezhi Mao, 2019). The mentioned paper works on three different datasets, namely, SemEval-2010 task 8, Casual-TimeBank, and Event StoryLine Data. However, in our replication of this project, we have only used SemEval-2010 task 8.

## 2.1 K-CNN Architecture

In this section, we will discuss the architecture of Knowledge-CNN. As mentioned, K-CNN combines human prior knowledge and information learned from state-of-the-art methods to extract relationships from natural language text. In conventional CNN, the weights of the convolutional filters are trained via back-propagation. However, in the case of knowledge-oriented channels, the filters are automatically generated using lexical knowledge

[2] Relation Extraction: Perspective from Convolutional Neural Networks

[3] SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals

[4] https://www.sciencedirect.com/science/article/abs/pii/S0957417418305177#!

bases such as FrameNet and WordNet, called word filters. The weights of the word filter in the known channel are equal to pre-trained word embeddings and thus considered constant. This significantly reduces the number of free parameters of the model; hence, the overfitting issue can be alleviated and training efficiency can be improved. Furthermore, word filter selection and clustering techniques can be introduced to remove redundant and non-discriminative features.

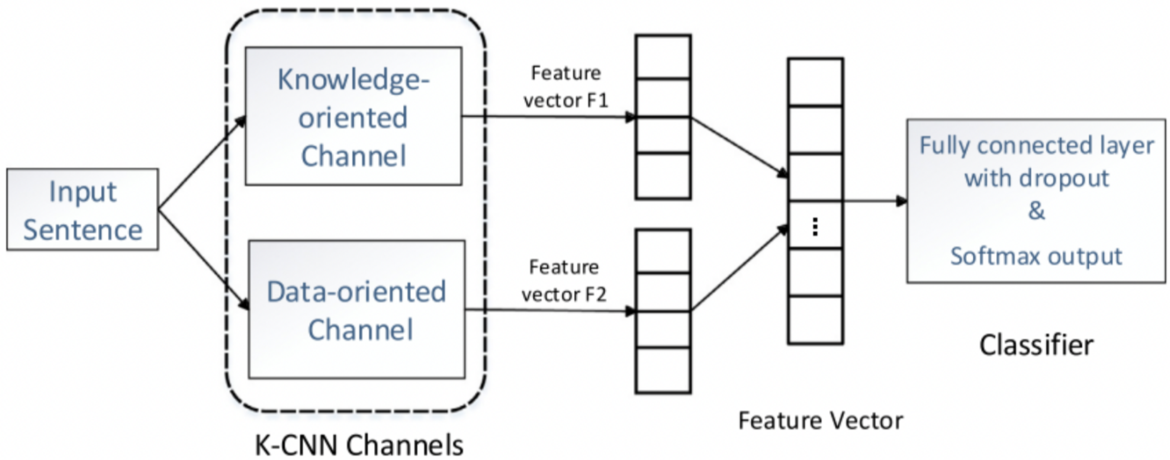


Figure 1: Overall Architecture of K-CNN (Pengfei Li Kezhi Mao, 2019)

### 2.1.1 Knowledge-Oriented channel

The K-CNN model successfully extracts keywords and cue phrases indicating causal relationships from sentences using automatic word filter bank generation. Thus, the convolutional filters are generated automatically. The Word Filter Bank Generation automatically creates CNN convolutional filters for the extraction of causal relationships without the need to train the model using a significant quantity of data. The filters are essentially causative word embeddings that have been taken from WordNet and FrameNet, two lexical knowledge libraries that are freely accessible.

**WordNet**: To represent various meanings or concepts, WordNet, a huge lexical database, arranges English words into sets of synonyms (referred to as synsets) (Fellbaum, 2010). The synsets are linked together in a hierarchical framework to represent their lexical and semantic relationships, and a gloss that includes instances explains each synset's meaning.

**FrameNet:** A lexical resource called FrameNet organizes English words and phrases into higher-level semantic frames that describe various concepts. It is based on the frame semantic theory (Ruppenhofer, Ellsworth, Petruck, Johnson, & Scheffczyk, 2016).
Each frame is a conceptual framework that describes a certain kind of action, relationship, or object. Each frame includes a conceptual definition, participants (also known as frame components), lexical units (a group of words that often appear in the frame), and connections to other frames.

### 2.1.2 Data-Oriented channel

The data-focused K-CNN channel employs traditional CNN to extract key characteristics of causal links from the training set. By using larger convolutional window lengths compared to knowledge channels, it captures longer dependencies throughout the sentence.
By allowing the convolutional filters to modify their weights based on the training data, we give the model the ability to learn such long-term dependencies and significant information that is overlooked by the knowledge-oriented channel. As a result, the data-oriented channel complements the knowledge-oriented channel, and the two channels working together enable K-CNN to effectively infer causal linkages from phrases.

**Sentence Representation:** It has been found in general observation that the majority of the keywords and the cue phrases such as “this is because”, “cause ”, “lead to” appear far away from two target entities.
Unlike the example from Section:1.1, “The financial <e1>*crisis*</e1> **resulted in** 12% <e2>*unemployment*<e2/> in this country”. (Causal), the maximum number (n1) of words in the following sentence is very large.
“In spite of great <e1>*effort*<e1/> put by the <e2>*government*<e2/> in improving more employment opportunities for low incomes, the financial crisis still **results in** 12% unemployment in this country ” (non-causal).
For both the sentences CNN may classify the relationship between e1 and e2 to be causal.
To avoid such ambiguity, position embeddings were proposed by Zeng et al. (2014) to deal with semantic analysis in long sentences. Furthermore, each input sentence is represented as a set of tokens (Figure 2) and converted into its lower case and base form using WordNet lemmatizer to eliminate morphological variances and make them compatible with WordNet tokens. In order to capture the syntactic and semantic information of tokens, each word is represented by a vector $w \in R^{e}$ by offering the word embedding table $W \in R^{e \times |V|}$ that is pre-trained using a large corpus, where e is the dimensionality of word embedding vectors and |V| is the vocabulary size. Since CNN works with fixed length inputs, the number of words in sentences (n1) is fixed.
The sentences with fewer words than n1 tokens are padded with zero embedding vectors. Hence, the input $x_k = \{x_1, x_2, ..., x_{n1}\}$ is represented as a sequence of real-value vector $emb_k = \{w_1, w_2, ..., w_{n1}\}$.

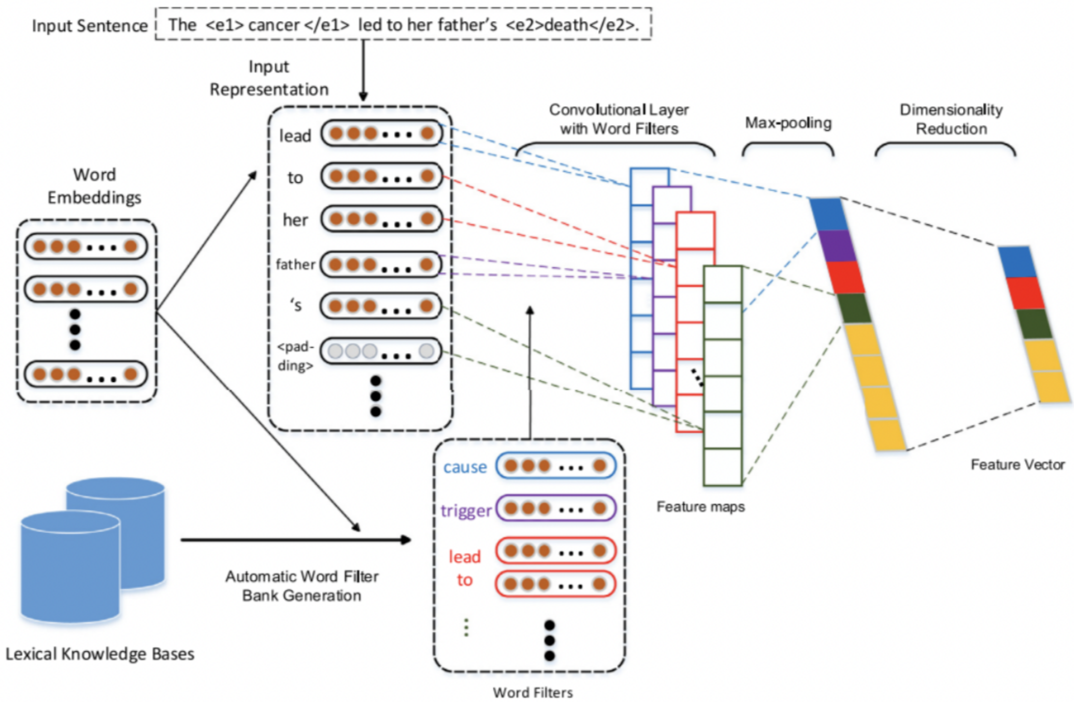


Figure 2: Introduction of Knowledge channel to CNN (Pengfei Li Kezhi Mao, 2019)

# 3. Approach and System replication

## 3.1 Challenges

Unlike CATENA, in the K-CNN model, things were to be built from scratch because of the unavailability of base code, and broken links for embeddings and dataset.

To get better ideas for the implementation, several papers were referred to. For a detailed understanding of the data-oriented model Nguyen & Grishman,2015; Santos, Xiang, & Zhou, 2015, for relation classification via deep learning, Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou and for [5]multi-way classification of Semantic Relations between pairs.

Due to the unavailability of the resources, initially, a lot of attempts were made to progress the project. After some failed attempts, we opted to build a relation classification model by following the related workflow from the parent paper (multi-way classification of Semantic Relations between pairs).

I have been able to perform data analysis, preprocess the SemEval-2010 task 8 dataset and build a neural network for the multi-way classification of the relations. However, I could not introduce a knowledge-oriented channel given the scarcity of time.

[5] Refrence: https://github.com/sahitya0000/Relation-Classification

## 3.2 Dataset analysis

Simple causal links are the main focus of the SemEval-2010 task 8 dataset. These connections specifically addressed the relationship with cue words like "cause," "result in," and other similar expressions.
Data set is composed of:
10717 annotated sample → (e1,e2) and relationship

- 478 Cause-Effect(e1,e2)
- 853 Cause-Effect(e2,e1)
- 9386 other samples

Moreover following entity relationships were analyzed during data analysis and preprocessing.

```
Message-Topic(e1,e2) 1
Message-Topic(e2,e1) 2
Product-Producer(e1,e2) 3
Product-Producer(e2,e1) 4
Instrument-Agency(e1,e2) 5
Instrument-Agency(e2,e1) 6
Entity-Destination(e1,e2) 7
Entity-Destination(e2,e1) 8
Cause-Effect(e1,e2) 9
Cause-Effect(e2,e1) 10
Component-Whole(e1,e2) 11
Component-Whole(e2,e1) 12
Entity-Origin(e1,e2) 13
Entity-Origin(e2,e1) 14
Member-Collection(e1,e2) 15
Member-Collection(e2,e1) 16
Content-Container(e1,e2) 17
Content-Container(e2,e1) 18
```

It can be observed, SemEval-2010 task 8 dataset has 9 directional relations, total, $9 \times 2 = 18$ +1 for other classes.

## 3.3 Project storyline and outputs

### 3.3.1 Pre-processing of data

A. Creating training-test attention
   In this section we performed pre-processing on the train and test data, which was already divided in the given dataset.
   - Cleaning of data (removal of extra space and special characters)
   - Tokenization
   - Converting tokens to lowercase
   - POS tagging
   - Embeddings: Word2Vec Pre-trained Embeddings

The processed training and testing data were stored in separate files for further processing.

B. Data-split: The data set is already divided into train and test data. We have randomly extracted the 792 validation data samples from the given training data of 8000 samples.
    - Training and Validation Split- train:validation = 9:1

```
1       "The system as described above has its greatest application in an arrayed <e1>configuration</e1> of antenna
<e2>elements</e2>."
Component-Whole(e2,e1)
Comment: Not a collection: there is structure here, organisation.

2       "The <e1>child</e1> was carefully wrapped and bound into the <e2>cradle</e2> by means of a cord."
Other
Comment:

3       "The <e1>author</e1> of a keygen uses a <e2>disassembler</e2> to look at the raw assembly code."
Instrument-Agency(e2,e1)
Comment:

4       "A misty <e1>ridge</e1> uprises from the <e2>surge</e2>."
Other
Comment:

5       "The <e1>student</e1> <e2>association</e2> is the voice of the undergraduate student population of the State
University of New York at Buffalo."
```

Figure 3: training data set before cleaning

```
1 Component-Whole(e2,e1) the system as described above has its greatest application in an arrayed e1_start configuration
e1_end of antenna e2_start elements e2_end .
2 Other the e1_start child e1_end was carefully wrapped and bound into the e2_start cradle e2_end by means of a cord .
3 Instrument-Agency(e2,e1) the e1_start author e1_end of a keygen uses a e2_start disassembler e2_end to look at the raw
assembly code .
6 Other this is the sprawling e1_start complex e1_end that is peru 's largest e2_start producer e2_end of silver .
7 Cause-Effect(e2,e1) the current view is that the chronic e1_start inflammation e1_end in the distal part of the stomach
caused by helicobacter pylori e2_start infection e2_end results in an increased acid production from the non-infected
upper corpus region of the stomach .
8 Entity-Destination(e1,e2) e1_start people e1_end have been moving back into e2_start downtown e2_end .
9 Content-Container(e1,e2) the e1_start lawsonite e1_end was contained in a e2_start platinum crucible e2_end and the
counter-weight was a plastic crucible with metal pieces .
```

Figure 4: clean training data set

C. Introducing pre-trained embeddings

    We have downloaded and extracted GoogleNews-vectors in our code and converted the words into float points to construct embedding vectors.

```
-0.00022588433600012248 -0.001012908721666691 -0.010476977797999708 0.03146318709199803 0.0019491069413334003
-0.04835933371966381 0.008710553770332682 -0.041882874847328416 0.02514677162166702 0.045829922259662016
0.0012610101896665773 -0.001755135828666413 -0.02505967693999945 0.042076889638327945 -0.04062638724566258
0.029448116114998535 0.018045934691997056 0.03598551912932831 -0.02661335783000228 -0.007926055424333635
0.027085851171999107 0.004249004574000113 0.052717857174669105 0.019318419361666797 0.005107683717332928
-0.027622642094333818 -0.033870997900995584 0.04400014751966207 0.013241271518665823 -0.04492411003766484
-0.030529173601666435 -0.04278228080632902 0.004646900222333109 -0.02742666811566531 0.003948020522667164
-0.07113646368865112 0.03723364946466072 0.007205472983999353 0.003087900879999821 0.008449408526666268
-0.008023713991333373 -0.02384378022666662 0.04025884059200218 0.05377640890765985 -0.046401671917997864
-0.07931488883863048 -0.04395082207465984 -0.016293921530329335 0.0079504759310002 0.04176778839832646
-0.06895921010965775 0.03716209873199277 -0.0013370641593334557 -0.003446856877331617 -0.02544627126333408
0.04805470252666673 -0.05851519298433866 -0.028776991084999006 -0.02021331286433288 -0.06874522918864537
-0.04953822839565935 0.045638062378664156 -0.05194669631299753 0.004481893263999823 6 0.007939646180665864
```

## 3.3. 2 Modeling of Neural Network: Data-oriented Channel

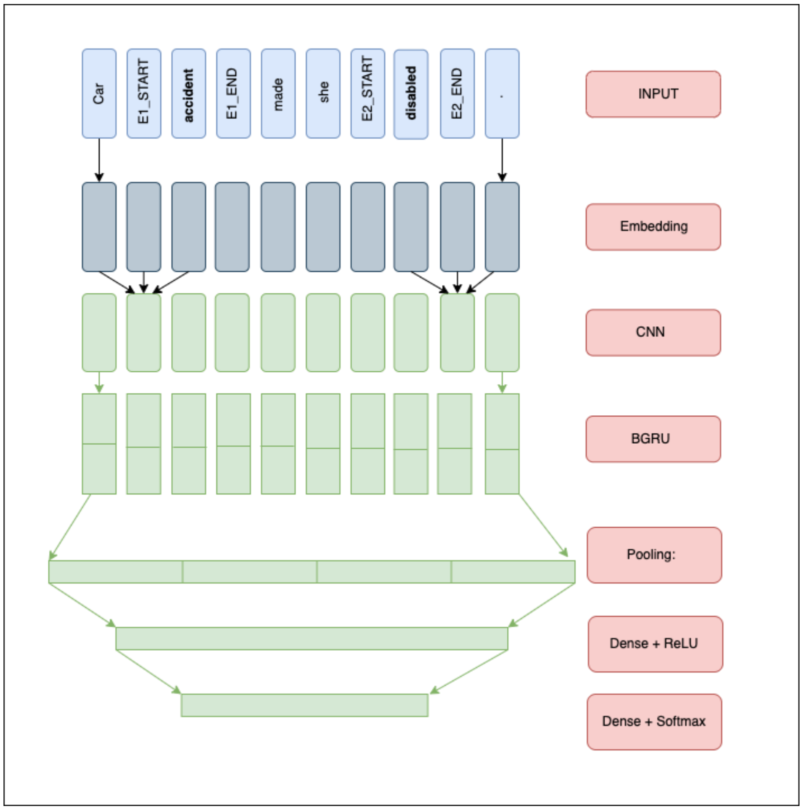


Figure 5: Model Architecture

In the data-oriented channel, CNN + Bidirectional Gated Recurrent Unit (BGRU) is modelled using the following parameters.

```
units 64
batch_size 128
dropout_emb 0.64
dropout_model 0.32
dropout_pen 0.32
l2_val 1e-05
learning_rate 1.0
activation_fn tanh
nb_epoch 256
adadelta <keras.optimizer_v2.adadelta.Adadelta object at 0x7f0976499bd0>
save_model True
es_epoch_stop 20

/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adadelta.py:74: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adadelta, self).__init__(name, **kwargs)
```

Figure 6: Hyperparameters

Model: "model_2"

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| words_input (InputLayer) | [(None, 101)] | 0 | [] |
| words_Embedding (Embedding) | (None, 101, 300) | 7035000 | ['words_input[0][0]'] |
| dropout_6 (Dropout) | (None, 101, 300) | 0 | ['words_Embedding[0][0]'] |
| conv1d_2 (Conv1D) | (None, 101, 256) | 230656 | ['dropout_6[0][0]'] |
| dropout_7 (Dropout) | (None, 101, 256) | 0 | ['conv1d_2[0][0]'] |
| bidirectional_2 (Bidirectional) | (None, 101, 128) | 123648 | ['dropout_7[0][0]'] |
| activation_4 (Activation) | (None, 101, 128) | 0 | ['bidirectional_2[0][0]'] |
| dense_6 (Dense) | (None, 101, 1) | 129 | ['activation_4[0][0]'] |
| permute_2 (Permute) | (None, 1, 101) | 0 | ['dense_6[0][0]'] |
| attn_softmax (Activation) | (None, 1, 101) | 0 | ['permute_2[0][0]'] |
| words_input_mask (InputLayer) | [(None, 101)] | 0 | [] |
| lambda_2 (Lambda) | (None, 1, 128) | 0 | ['attn_softmax[0][0]', 'activation_4[0][0]'] |
| global_max_pooling1d_2 (GlobalMaxPooling1D) | (None, 128) | 0 | ['activation_4[0][0]'] |
| mask_max_pooling_layer_2 (MaskMaxPoolingLayer) | (None, 128) | 0 | ['activation_4[0][0]', 'words_input_mask[0][0]'] |
| flatten_2 (Flatten) | (None, 128) | 0 | ['lambda_2[0][0]'] |
| concatenate_2 (Concatenate) | (None, 384) | 0 | ['global_max_pooling1d_2[0][0]', 'mask_max_pooling_layer_2[0][0]', 'flatten_2[0][0]'] |
| dropout_8 (Dropout) | (None, 384) | 0 | ['concatenate_2[0][0]'] |
| dense_7 (Dense) | (None, 300) | 115500 | ['dropout_8[0][0]'] |
| dense_8 (Dense) | (None, 19) | 5719 | ['dense_7[0][0]'] |
| activation_5 (Activation) | (None, 19) | 0 | ['dense_8[0][0]'] |

Figure 7: Layers

### 3.3.3 Future work: Knowledge-Oriented Channel

This section of the project is for future work on this project to evaluate and compare the results.

## 3.4 Results

Confusion matrix taking directionality into account:

```
<<< (2*9+1)-WAY EVALUATION (USING DIRECTIONALITY)>>>:

Confusion matrix:
        C-E1 C-E2 C-W1 C-W2 C-C1 C-C2 E-D1 E-D2 E-O1 E-O2 I-A1 I-A2 M-C1 M-C2 M-T1 M-T2 P-P1 P-P2  _O_ <-- classified as
      +-----------------------------------------------------------------------------------------------+ -SUM- skip ACTUAL
 C-E1 | 127    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    5 |  134    0  134
 C-E2 |   0  174    0    0    0    0    0    0    9    0    0    0    0    0    0    0    4    0    7 |  194    0  194
 C-W1 |   0    1  127    3    3    0    1    0    0    0    1    2    3    4    0    0    0    0   17 |  162    0  162
 C-W2 |   0    0    1  115    0    1    1    0    1    0    0    6    0    6    5    0    0    2   12 |  150    0  150
 C-C1 |   0    0    0    0  144    2    3    0    0    0    0    0    0    0    0    0    0    0    4 |  153    0  153
 C-C2 |   0    0    0    0    2   30    0    0    0    1    0    0    0    1    0    0    0    0    5 |   39    0   39
 E-D1 |   0    0    2    0   12    0  262    0    0    0    0    1    0    0    1    0    0    0   13 |  291    0  291
 E-D2 |   0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0 |    1    0    1
 E-O1 |   0    0    0    0    2    0    2    0  193    1    0    1    0    0    1    2    2    0    7 |  211    0  211
 E-O2 |   1    0    0    0    1    0    0    0    0   40    1    0    0    0    0    0    0    0    4 |   47    0   47
 I-A1 |   0    0    0    0    0    0    0    0    0    0   15    2    0    0    0    0    2    0    3 |   22    0   22
 I-A2 |   0    0    0    5    0    0    1    0    0    0    0  105    0    0    0    0    0    3   20 |  134    0  134
 M-C1 |   0    0    1    0    0    0    1    0    1    0    0    0   22    2    0    0    0    0    5 |   32    0   32
 M-C2 |   0    0    1    2    0    0    0    0    0    0    0    0    0  183    0    0    1    0   14 |  201    0  201
 M-T1 |   0    0    0    1    0    0    0    0    1    0    0    0    0    1  188    4    0    0   15 |  210    0  210
 M-T2 |   0    0    0    0    0    0    0    0    1    0    0    0    0    0    1   46    0    0    3 |   51    0   51
 P-P1 |   0    3    1    0    0    0    0    0    4    0    2    0    0    0    1    0   95    1    1 |  108    0  108
 P-P2 |   1    0    0    3    0    0    2    0    1    1    0    9    1    1    1    2    0   83   18 |  123    0  123
  _O_ |   6    8    9   20   25    3   17    0   31    7    2   18    9   23   21    8   13    8  226 |  454    0  454
      +-----------------------------------------------------------------------------------------------+
 -SUM-  135  187  142  149  189   36  291    0  242   51   21  144   35  221  219   62  117   97  379   2717    0 2717

Coverage = 2717/2717 = 100.00%
Accuracy (calculated for the above confusion matrix) = 2175/2717 = 80.05%
Accuracy (considering all skipped examples as Wrong) = 2175/2717 = 80.05%
Accuracy (considering all skipped examples as Other) = 2175/2717 = 80.05%
```

```
Results for the individual relations:
      Cause-Effect(e1,e2) :    P =  127/ 135 =  94.07%     R =  127/ 134 =  94.78%     F1 =  94.42%
      Cause-Effect(e2,e1) :    P =  174/ 187 =  93.05%     R =  174/ 194 =  89.69%     F1 =  91.34%
   Component-Whole(e1,e2) :    P =  127/ 142 =  89.44%     R =  127/ 162 =  78.40%     F1 =  83.55%
   Component-Whole(e2,e1) :    P =  115/ 149 =  77.18%     R =  115/ 150 =  76.67%     F1 =  76.92%
 Content-Container(e1,e2) :    P =  144/ 189 =  76.19%     R =  144/ 153 =  94.12%     F1 =  84.21%
 Content-Container(e2,e1) :    P =   30/  36 =  83.33%     R =   30/  39 =  76.92%     F1 =  80.00%
Entity-Destination(e1,e2) :    P =  262/ 291 =  90.03%     R =  262/ 291 =  90.03%     F1 =  90.03%
Entity-Destination(e2,e1) :    P =    0/   0 =   0.00%     R =    0/   1 =   0.00%     F1 =   0.00%
     Entity-Origin(e1,e2) :    P =  193/ 242 =  79.75%     R =  193/ 211 =  91.47%     F1 =  85.21%
     Entity-Origin(e2,e1) :    P =   40/  51 =  78.43%     R =   40/  47 =  85.11%     F1 =  81.63%
 Instrument-Agency(e1,e2) :    P =   15/  21 =  71.43%     R =   15/  22 =  68.18%     F1 =  69.77%
 Instrument-Agency(e2,e1) :    P =  105/ 144 =  72.92%     R =  105/ 134 =  78.36%     F1 =  75.54%
 Member-Collection(e1,e2) :    P =   22/  35 =  62.86%     R =   22/  32 =  68.75%     F1 =  65.67%
 Member-Collection(e2,e1) :    P =  183/ 221 =  82.81%     R =  183/ 201 =  91.04%     F1 =  86.73%
     Message-Topic(e1,e2) :    P =  188/ 219 =  85.84%     R =  188/ 210 =  89.52%     F1 =  87.65%
     Message-Topic(e2,e1) :    P =   46/  62 =  74.19%     R =   46/  51 =  90.20%     F1 =  81.42%
  Product-Producer(e1,e2) :    P =   95/ 117 =  81.20%     R =   95/ 108 =  87.96%     F1 =  84.44%
  Product-Producer(e2,e1) :    P =   83/  97 =  85.57%     R =   83/ 123 =  67.48%     F1 =  75.45%
                   _Other :    P =  226/ 379 =  59.63%     R =  226/ 454 =  49.78%     F1 =  54.26%

Micro-averaged result (excluding Other):
P = 1949/2338 =  83.36%     R = 1949/2263 =  86.12%     F1 =  84.72%

MACRO-averaged result (excluding Other):
P =  76.57%     R =  78.82%     F1 =  77.44%
```

Confusion matrix without using directionality

```
<<< (9+1)-WAY EVALUATION IGNORING DIRECTIONALITY >>>:

Confusion matrix:
        C-E  C-W  C-C  E-D  E-O  I-A  M-C  M-T  P-P  _O_ <-- classified as
      +--------------------------------------------------+ -SUM- skip ACTUAL
  C-E | 302    0    0    0   10    0    0    0    4   12 |  328    0  328
  C-W |   1  246    4    2    1    9   13    5    2   29 |  312    0  312
  C-C |   0    0  178    3    1    0    1    0    0    9 |  192    0  192
  E-D |   0    2   12  263    0    1    0    1    0   13 |  292    0  292
  E-O |   1    0    3    2  234    2    0    3    2   11 |  258    0  258
  I-A |   0    5    0    1    0  122    0    0    5   23 |  156    0  156
  M-C |   0    4    0    1    1    0  207    0    1   19 |  233    0  233
  M-T |   0    1    0    0    2    0    1  239    0   18 |  261    0  261
  P-P |   4    4    0    2    6   11    2    4  179   19 |  231    0  231
  _O_ |  14   29   28   17   38   20   32   29   21  226 |  454    0  454
      +--------------------------------------------------+
 -SUM-  322  291  225  291  293  165  256  281  214  379   2717    0 2717

Coverage = 2717/2717 = 100.00%
Accuracy (calculated for the above confusion matrix) = 2196/2717 = 80.82%
Accuracy (considering all skipped examples as Wrong) = 2196/2717 = 80.82%
Accuracy (considering all skipped examples as Other) = 2196/2717 = 80.82%
```

```
Results for the individual relations:
             Cause-Effect :    P =  302/ 322 =  93.79%     R =  302/ 328 =  92.07%     F1 =  92.92%
          Component-Whole :    P =  246/ 291 =  84.54%     R =  246/ 312 =  78.85%     F1 =  81.59%
        Content-Container :    P =  178/ 225 =  79.11%     R =  178/ 192 =  92.71%     F1 =  85.37%
       Entity-Destination :    P =  263/ 291 =  90.38%     R =  263/ 292 =  90.07%     F1 =  90.22%
            Entity-Origin :    P =  234/ 293 =  79.86%     R =  234/ 258 =  90.70%     F1 =  84.94%
        Instrument-Agency :    P =  122/ 165 =  73.94%     R =  122/ 156 =  78.21%     F1 =  76.01%
        Member-Collection :    P =  207/ 256 =  80.86%     R =  207/ 233 =  88.84%     F1 =  84.66%
            Message-Topic :    P =  239/ 281 =  85.05%     R =  239/ 261 =  91.57%     F1 =  88.19%
         Product-Producer :    P =  179/ 214 =  83.64%     R =  179/ 231 =  77.49%     F1 =  80.45%
                   _Other :    P =  226/ 379 =  59.63%     R =  226/ 454 =  49.78%     F1 =  54.26%

Micro-averaged result (excluding Other):
P = 1970/2338 =  84.26%     R = 1970/2263 =  87.05%     F1 =  85.63%

MACRO-averaged result (excluding Other):
P =  83.46%     R =  86.72%     F1 =  84.93%
```

Final Results:

```
9+1)-WAY EVALUATION TAKING DIRECTIONALITY INTO ACCOUNT -- OFFICIAL >>>:

sion matrix:
    C-E  C-W  C-C  E-D  E-O  I-A  M-C  M-T  P-P  _O_ <-- classified as
 +--------------------------------------------------+ -SUM- xDIRx skip  ACTUAL
 | 301    0    0    0   10    0    0    0    4   12 |  327     1     0    328
 |   1  242    4    2    1    9   13    5    2   29 |  308     4     0    312
 |   0    0  174    3    1    0    1    0    0    9 |  188     4     0    192
 |   0    2   12  262    0    1    0    1    0   13 |  291     1     0    292
 |   1    0    3    2  233    2    0    3    2   11 |  257     1     0    258
 |   0    5    0    1    0  120    0    0    5   23 |  154     2     0    156
 |   0    4    0    1    1    0  205    0    1   19 |  231     2     0    233
 |   0    1    0    0    2    0    1  234    0   18 |  256     5     0    261
 |   4    4    0    2    6   11    2    4  178   19 |  230     1     0    231
 |  14   29   28   17   38   20   32   29   21  226 |  454     0     0    454
 +--------------------------------------------------+
-  321  287  221  290  292  163  254  276  213  379   2696    21     0   2717

age = 2717/2717 = 100.00%
acy (calculated for the above confusion matrix) = 2175/2717 = 80.05%
acy (considering all skipped examples as Wrong) = 2175/2717 = 80.05%
acy (considering all skipped examples as Other) = 2175/2717 = 80.05%

ts for the individual relations:
        Cause-Effect :    P =  301/( 321 +   1) =  93.48%     R =  301/ 328 =  91.77%     F1 =  92.62%
     Component-Whole :    P =  242/( 287 +   4) =  83.16%     R =  242/ 312 =  77.56%     F1 =  80.27%
   Content-Container :    P =  174/( 221 +   4) =  77.33%     R =  174/ 192 =  90.62%     F1 =  83.45%
  Entity-Destination :    P =  262/( 290 +   1) =  90.03%     R =  262/ 292 =  89.73%     F1 =  89.88%
       Entity-Origin :    P =  233/( 292 +   1) =  79.52%     R =  233/ 258 =  90.31%     F1 =  84.57%
   Instrument-Agency :    P =  120/( 163 +   2) =  72.73%     R =  120/ 156 =  76.92%     F1 =  74.77%
   Member-Collection :    P =  205/( 254 +   2) =  80.08%     R =  205/ 233 =  87.98%     F1 =  83.84%
       Message-Topic :    P =  234/( 276 +   5) =  83.27%     R =  234/ 261 =  89.66%     F1 =  86.35%
    Product-Producer :    P =  178/( 213 +   1) =  83.18%     R =  178/ 231 =  77.06%     F1 =  80.00%
              _Other :    P =  226/( 379 +   0) =  59.63%     R =  226/ 454 =  49.78%     F1 =  54.26%

-averaged result (excluding Other):
949/2338 =  83.36%     R = 1949/2263 =  86.12%     F1 =  84.72%


he official score is (9+1)-way evaluation with directionality taken into account: micro-averaged F1 = 84.72%
```

# 4. Conclusion

We studied Knowledge-oriented CNN (Pengfei Li and Kezhi Mao, 2019) and understood the architecture and solutions it provides to the pre-existing works on the topic.
Although we were not able to replicate the parent paper for this project, the different approaches were followed by referring to a plethora of papers. One such approach is used to construct a data-oriented channel for the attempt to replicate the results (Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, Stan Szpakowicz, 2010). Our neural network model is composed of a Convolutional neural network and a Bidirectional Gated Recurrent Unit with the combination of different pooling layers (max, entity and attention pool). The resulting model provides decent results, though not as good as the K-CNN approach.

**Acknowledgements:** The project is based on ongoing research on event detection in natural language processing. K-CCN is the outcome of one of such methods and is progressed under the supervision of Prof. Raphael Troncy, Pasquale Lisena and Youssra Rebboud.

# 5. References

[1] Pengfei Li, Kezhi Mao (2019). Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts.

[2] Nguyen, T. H., & Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In Proceedings of the first workshop on vector space modelling for natural language processing (pp. 39–48).

[3] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, Stan Szpakowicz (2010). SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals

[4] Santos, C. N. D., Xiang, B., & Zhou, B. (2015). Classifying relations by ranking with convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 626–634).

[5] Ackerman, E. J. M. (2012). Extracting a causal network of news topics. In Proceedings of the OTM confederated international conferences "on the move to meaningful internet systems" (pp. 33–42). Springer.

[6] David, M. L. (2018). Online statistics education: An interactive multimedia course of study. http://onlinestatbook.com.
Fellbaum, C. (2010). Theory and applications of ontology: Computer applications. Wordnet.

[7] Girju, R., Beamer, B., Rozovskaya, A., Fister, A., & Bhat, S. (2010). A knowledge-rich approach to identifying semantic relations between nominals. Information Processing & Management, 46(5), 589–610.

[8] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning: 1. MIT press Cambridge.

[9] Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., et al. (2009). SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the workshop on semantic evaluations: Recent achievements and future directions (pp. 94–99). Association for Computational Linguistics.

[10] Khoo, C. S., Kornfilt, J., Oddy, R. N., & Myaeng, S. H. (1998). Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. Literary and Linguistic Computing, 13(4), 177–186.

[11] Lee, J. Y., Dernoncourt, F., & Szolovits, P. (2017). Mit at semeval-2017 task 10: Relation extraction with convolutional neural networks. In Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017) (pp. 978–984). Association for Computational Linguistics.

[12] Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In Proceedings of the COLING twenty-fifth international conference on computational linguistics: technical papers (pp. 2335–2344).

[13] Jie Yang, Soyeon Caren Han, Josiah Poon, (2021) A Survey on Extraction of Causal Relations from Natural Language Text