

ssn_nlp@FIRE2020 : Automatic extraction of causal relations using deep learning and machine translation approaches

Thenmozhi D, Arunima S, Amlan Sengupta, Avantika Balaji

Department of CSE, SSN College of Engineering, Chennai

theni_d@ssn.edu.in

{arunima17016, amlan17008, avantika17021}@cse.ssn.edu.in

Abstract

Causality can be understood as the relationship between two events such that the occurrence of one event results in the occurrence of the other event either directly or indirectly. This paper aims to identify whether the given sentences have a causality effect present in them and to classify the cause and effect words/phrases if present. The approach used for classification uses deep learning algorithms and the annotation task uses machine translation. These models are applied to the dataset provided by CEREX@FIRE2020. The best result for the causality identification part was obtained from BiLSTM with an F1 score of 0.60 and for the second task of annotation as cause and effect, NMT with Bahdanau attention mechanism with an F1 score of 0.44.

Keywords

CEREX, Cause, Effect, Causal connective, Logistic Regression, Bi-LSTM, NMT

1. Introduction

Causality is defined as the relation between a cause and its effect. A cause is why an event happens. An effect is an event that happens because of the cause. Any sentence with a causal expression has the following three components: cause, effect, and causal connective. For example, in the sentence : “Due to inflation, the dollar is worth less than before”, - the event “inflation” is the cause and the event “the dollar is worth less than before” is the effect. “Due to” is the causal connective here.

In recent times, automatic extraction of semantic relations, in particular, automatic extraction of causal relations, has become essential for many natural language processing (NLP) applications like question answering, document-summarization, opinion mining, event analysis. One of the easiest ways to express cause-effect relations is through the form “A causes B” or “B is caused by A”. But causality can be expressed through a wide variety of syntactic expressions, and these variations are hard to represent using a single model. Therefore, due to the presence of complex grammatical structures in sentences, the automatic extraction of causal relations becomes a hard NLP problem to solve. The task here is two-fold.¹ The first task is to identify whether a given sentence contains a causal event (cause/effect). Two models were employed for this task - Logistic Regression and Bidirectional Long short term memory (Bi-LSTM). The second task is to annotate each word in a sentence in terms of four labels (C - cause, E- effect, CC- Causal Connective, and None). This task was implemented using the NMT model.

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

ORCID:



© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹<https://sites.google.com/view/cerex-fire2020/home>

2. Related Works

Several research work has been reported in the field of automatic extraction of cause and effect from natural language text. The authors of [1] have proposed a linguistically informed recursive neural network architecture for automatic extraction of cause-effect relations from text. The proposed architecture uses word level embeddings and other linguistic features to detect causal events and their effects mentioned within a sentence.

Cause-effect relations from documents in Metallurgy and Materials Science are extracted in this paper[2]. They used a BiLSTM model to annotate each word for their dataset which was created using distant supervision. LSTM based binary classifier was used for predicting whether any sentence expresses causality or not.

The plausible cause-effect pairs are identified through a set of logical rules based on dependencies between words. Then Bayesian inference is used to reduce the number of pairs produced by ambiguous patterns[3].

The author[4] extracts the causal knowledge from a medical database using graphical patterns. The sentences were parsed using Conexor's Functional Dependency Grammar (FDG) of English parser, which is used to generate a representation of the syntactic structure of the sentence that is the parse tree. The information extraction process included the matching of causality patterns with the parse trees of the sentences, both of which were represented in the linear conceptual graph notation.

Roxana Girjud uses explicit intra-sentential patterns where the verb is a simple causative. A transitive relation between verb synsets is known as the CAUSE-TO relation. WordNet consists of numerous causation relationships between nouns which are always true. One way to determine such relationships is to look for all patterns that occur between a noun entry and another noun in the corresponding gloss definition. This is the basis for the detection of Causal Relations for Question Answering mentioned in the paper[5].

The approach used by Blanco for the detection and extraction of causations is based on the use of syntactic patterns that may encode causation. They[6] then redefine the problem as a classification between two classes: encoding or not encoding causation (cause or -cause). The model used an implementation of Bagging with C4.5 decision trees.

3. Data Analysis and Pre-processing

The training data for both the tasks released by CEREX 2020 included 6000 rows containing 4 columns, the sno, sentence, cause, and effect. The sno is unique for every row, sentence consists of the entire sentence, cause is the causal part of the sentence and effect is the effect part of the sentence.

For Task A, the test dataset consisted of 764 sentences having 2 columns i.e. the sno and sentence. And the test dataset for Task B contained 178 rows and two columns which were Sent_id i.e the sentence and id and sentence. The sno and Sent_id are distinct for every sentence.

The pre-processing done for Task A included removing emojis and adding an additional label column which consists of two values i.e. 0 and 1. A sentence is classified as not causal if it doesn't contain both cause and effect. These kinds of sentences are labeled as 0. Otherwise, if a sentence has both cause and effect or just has either one of them, it is causal and labeled as 1. The cause and effect columns are then dropped. Finally, the data is split in the ratio of 80%-20% and given to the Bi-LSTM model. For the Logistic Regression classifier, the data is split in the ratio of 70%-30% and given as the input.

For Task B, the data was preprocessed by removing emojis and punctuations with the help of regular expressions. Extra spaces in each sentence were removed. We created a list of common causal connectives, so along with the cause and effect words we can identify the causal connectives in the sentence too. The list contained both words and phrases. Using wordnet², synonyms of each causal connective were found and their past, present, future participle forms were added to the list.

Each word in the dataset was annotated by its specific label. If a word did not belong in either cause, effect or was not a causal connective it was labeled as O. For the training, the dataset was split into multiple files. 'train.in' consists of the first 4200 sentences from the dataset and 'train.out' consists of the respective labels of each word present in the 'train.in'. The labels were C(cause), E(effect), CC(causal connective), O(None). 'dev.in' consists of the remaining 1800 sentences from the dataset and 'dev.out' consists of the labels generated by the model for each word in 'dev.in'. 'vocab.in' contains all the distinct set of words present in the dataset and 'vocab.out' contains the distinct set of labels to be generated by the model i.e CC, C, E and O.

4. Methodology and Implementation

The general methodology which we have followed contains two steps i.e. model training and post-processing. For the first task, the input to the model consisted of the sentences with their respective labels, i.e. 0 or 1. For the second task, the preprocessed data was stored in the form of an input file i.e. train.in. The output file, train.out, consisted of the corresponding labels for each word in train.in. After the result is obtained from the model, postprocessing is performed. For the first task, the predicted labels along with their sentences are returned in a file. For the second task, the resultant data obtained is post-processed by the removal of extra spaces. This is followed by annotating each word with its respective label i.e C, E, and CC in the format word/label. If a word contains the label O then it is not annotated. The result is stored in a new CSV file.

4.1. TASK A :

In this task, our goal is to identify whether or not a sentence contains a causal event and classify the sentences as 1 (if the sentence contains a causal event) or 0 (if the sentence does not contain a causal event). Two approaches have been proposed for this task, which are Logistic Regression and Bi-LSTM.

Model	Accuracy	Precision	Recall	F1-Score
Bi-LSTM	0.92	0.94	0.95	0.95
LR	0.91	0.72	0.77	0.74

We have employed the Logistic Regression Classifier as it works well with small datasets. Since the data isn't evenly distributed across the two classes, we have performed oversampling for the minority class 0. This is done using SMOTE (Synthetic Minority Oversampling Technique)³ such that the ratio of the two classes is 1:1. Then, the Logistic Regression Classifier was used to classify sentences as having causal events or not. Our second approach was based on Bi-LSTM as it effectively increases the amount of information available to the network, improving the context available to the algorithm. It outperformed LR based on the training dataset.

²<https://wordnet.princeton.edu/>

³https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html

4.2. TASK B:

For this task, our aim is to annotate each word with its respective label which is CC, C, E or None. We have used three NMT⁴ models to do the same, i.e. NMT with Scaled Luong and Normed Bahdanau attention mechanisms, and NMT without an attention mechanism. In all three models, the recurrent unit is LSTM.

Model	Accuracy
NMT with Normed Bahdanau	22.90%
NMT with Scaled Luong	22.62%
NMT without attention mechanism	21.95%

We started with the NMT model without an attention mechanism. The input is encoded into a fixed dimensional vector and then is decoded into the target labels from the vector. For the Scaled Luong attention mechanism, the input file is encoded and reduced to attention scores using simple matrix multiplication which makes it faster and more space-efficient and is then decoded into the target labels present in train.out. Finally, we used Normed Bahdanau which performs a linear combination of encoder states and the decoder states. The model predicts a target word based on the context vectors associated with the initial position and the previously generated target words. We chose Normed Bahdanau as it works better with smaller datasets and achieved a higher accuracy on the train dataset compared to the other two models.

5. Results

We have evaluated our models using the test data of CEREX@FIRE2020 . The performance was analyzed using the metrics namely precision, recall and F1-Score. We secured the second place for the Binary Classification task. For the Tagging task, we outperformed the other teams and came first.

Team	Precision	Recall	F1-Score	Task
CSECU.DSG	0.51	0.91	0.65	A (Binary Classification)
ssn_nlp	0.46	0.87	0.60	
ssn_nlp	0.36	0.57	0.44	B (Tagging)
CSECU.DSG	0.32	0.51	0.39	

The submission made by ssn_nlp for Task A i.e classification of sentences as causal or not was the deep learning approach using BiLSTM. The model obtained F1-Score of 0.60 against the test set. A precision score of 0.46 and a recall score of 0.87 was encountered.

Our team ssn_nlp officially submitted an NMT model using Normed Bahdanau attention mechanism for Tak B i.e tagging each word with its respective label. The model performed on the test set with F1-Score of 0.44, a precision score of 0.36 and, a recall score of 0.57.

⁴<https://opennmt.net/>

6. Conclusion

We used a Bi-LSTM model for the classification of sentences as containing causal events or not. This performed better than the Logistic Regression Classifier. The model achieved an accuracy of 91% and an F1 score of 0.95. For annotating the sentences with their respective labels, NMT with Bahdanau attention mechanism was employed as it works better with smaller datasets compared to the Scaled-Luong attention mechanism that is generally used for larger datasets. It produced an accuracy of 0.44. Future developments would include constructing an enriched corpus of causal connectives and incorporating more linguistic features.

References

- [1] T. Dasgupta, R. Saha, L. Dey, A. Naskar, Automatic extraction of causal relations from text using linguistically informed deep neural networks, in: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, 2018, pp. 306–316.
- [2] S. Pawar, R. Sharma, G. K. Palshikar, P. Bhattacharyya, V. Varma, Cause–effect relation extraction from documents in metallurgy and materials science, Transactions of the Indian Institute of Metals 72 (2019) 2209–2217.
- [3] A. Sorgente, G. Vettigli, F. Mele, Automatic extraction of cause-effect relations in natural language text., DART@ AI* IA 2013 (2013) 37–48.
- [4] C. S. Khoo, S. Chan, Y. Niu, Extracting causal knowledge from a medical database using graphical patterns, in: Proceedings of the 38th annual meeting of the association for computational linguistics, 2000, pp. 336–343.
- [5] R. Girju, Automatic detection of causal relations for question answering, in: Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, 2003, pp. 76–83.
- [6] E. Blanco, N. Castell, D. I. Moldovan, Causal relation extraction., in: Lrec, volume 66, 2008, p. 74.