Causal Discovery from Natural Language Text using Context and Dependency Information

Shania Mitra[1†] and Arun K. Tangirala[2]

[1]Department of Chemical Engineering, Indian Institute of Technology Madras, India
(Tel: +91-72595-26778; E-mail: shaniamitra9@gmail.com)
[2]Department of Chemical Engineering, Indian Institute of Technology Madras, India
(Tel: +91-94447-43119; E-mail: arunkt@iitm.ac.in)

Abstract: Causality analysis is one of the emerging topics of importance in fault diagnosis, network reconstruction and in performance analysis of control systems. Almost all current day literature in the engineering arena makes use of numerical data. However, many processes generate insightful textual data that is left unanalysed. In this work we present a novel method for causal discovery from textual data using context and dependency information. The proposed BiLSTM and Node2Vec architecture uses word-level embeddings, linguistic features to gather contextual information and dependency-parse tree representations to gauge how entities in a sentence are related. The extracted phrases in each sentence are labelled as causes, effects or causal connectives. Our work is evaluated on datasets such as SemEval, BECAUSE 2.0 and CausalBank and compared against a baseline Conditional Random Field (CRF) based supervised BiLSTM model. We demonstrate that the presence of both context and dependency information significantly improves performance and also extend our analysis to operator logs, exhibiting its effectiveness in extracting causal events in process logs. The work hopes to generate immense value for the technology envisaged in Industry 4.0

Keywords: Causal Discovery, Natural Language Processing, Textual Data, BiLSTM, Node2Vec

1. INTRODUCTION

The challenge of inferring causal relationships between process variables has piqued the interest of scientists from numerous disciplines. Traditionally in engineering, however, most causal extraction has taken place on numerical data, with abundant literature [1-3] highlighting different methods for doing so. With the advent of new age technologies and the dawn of Industry 4.0, it is essential to extend the causal discovery capabilities to the world of textual data. The resulting techniques would be an attractive complement to those that work with numerical data. Processing of textual data generated by process operations is at a relatively nascent stage, leading to key insights and causal information being overlooked. The present work focuses on extracting cause and effect phrases from sentences in various corpora. This can further be extended to operator logs from various sub-parts of a process such that we can finally build a causal graph across the entire process allowing for efficient fault diagnosis and effect prediction.

Manual extraction of causes and effects from sentences is both time-consuming and laborious warranting the need for an automatic extraction mechanism. Automatic extraction can be performed by the use of either rule-based systems or machine learning approaches. Rule-based expert systems [4], [5] are often challenged by low accuracy since language is an intricate phenomenon with interaction between entities adding additional complexity. Each rule, being applicable to only a limited set of entities, makes it difficult to improve upon these systems.

Traditional machine learning algorithms [6], [7] rely primarily on precise feature engineering and may not be well suited to diverse types of datasets. Other machine learning-based methods frequently seek for single-word causes and effects, which are insufficient to capture causality in lengthy phrases. Yet others make use of a variety of word embeddings and linguistic features [8] to help make the architectures more informed. However, none of these take into account how different entities in a sentence relate to each other, limiting their performance in causal discovery.

In this work, we explore the use of Bidirectional LSTMs that are fed with linguistic features - features relating to individual words and dependency features - representations based on relations between entities, capturing the interaction between the words in a sentence, over and above regular word embeddings. The addition of context and dependency features helps in making the model more informed about the semantics of the sentence. With the use of such a linguistically informed deep architecture, we effectively extract cause and effect phrases from sentences without having to perform tedious feature engineering. Implementation of the current algorithm on two datasets demonstrate the superior performance as compared to other models.

The rest of this paper is organized as follows. Section 2 briefly reviews the concepts of Natural Language Processing (NLP) required to understand the work along with details of models and metrics used. Section 3 highlights the method used to extract causal relations from sentences. Section 4 studies the results when these models are applied to various datasets and finally the paper ends with a few concluding remarks in Section 5.

† Shania Mitra is the presenter of this paper.

2. BACKGROUND

This section provides some background information required to understand the working of the model.

2.1. Natural Language Processing

Natural Language Processing (NLP) is an area of Artificial Intelligence (AI) that enables machines to understand human language. This helps analyze and understand large volumes of unstructured data such as text in a short span of time.

2.2. Causal Discovery using NLP

Causal Discovery in textual data can be categorised into different tasks, some of which are as follows:

1. **Causal/Non-causal classification:** This task entails the classification of sentences into causal and non-causal. E.g.-1: Incorrect tuning of the controller led to oscillations in the output $(Causal)$ E.g.-2: An endothermic reaction requires heating $(Non-causal)$
2. **Relation extraction:** This task consists of identifying the relationship between two tagged entities in a sentence. E.g. Ram caught a *cold* due to the chilled *weather* (Relation type: Results From, i.e., the cold resulted from the weather)
3. **Intrasentential explicit causal tagging:** This task consists of extracting causality within a sentence with explicit causal connective. E.g. Financial stress is the leading cause of divorce (Cause: financial stress, Effect: Divorce, Causal connective: cause)
4. **Intrasentential implicit causal tagging:** This task consists of extracting causality within a sentence with no explicit causal connective. E.g. Financial stress can speed divorce up (Cause: financial stress, Effect: Divorce)
5. **Intersentential causal tagging:** Extracting causality from multiple sentences at a time comes under the purview of this task. E.g.. It is thought that heavy vehicles cause roads to crack. However, poor construction is the most important factor. (Cause: Heavy vehicles, poor construction Effect: Cracked road)

The present work focuses on the tasks of intrasentential implicit and explicit causal tagging.

2.3. Preprocessing Techniques

The models used in this work are fed with numerical and categorical features along with words from English sentences. To the models however, only numerical values are comprehensible. Thus, words and categorical features need to be converted to numerical features and the following methods are adopted for the same.

2.3.1. One-Hot Encoding

For categorical variables that do not have any ordinal relationship, we use One-Hot encoding introducing binary indicator variables for each category of the original feature, leading to a large sparse matrix that is understood by the model.

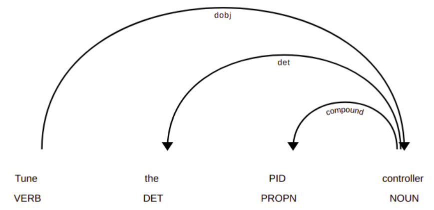

Fig. 1 Example of a Dependency Parse Tree

2.3.2. GloVE Embeddings

GloVe [9] is an unsupervised learning algorithm for obtaining vectors for words, representative of their meaning, by taking into account local information such as surrounding words and global information such as a word-word co-occurence matrix. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. In this work, every word in a each sentence of the corpus is converted into a 50-dimensional vector given by GloVe, making the input comprehensible to the model.

2.4. Dependency Parse Trees

A dependency parse tree is a graph $G(E, V)$ where the set of vertices contains the words in the sentence and each edge in it connects two words. Each edge in E has a type which defines the grammatical relation that occurs between two words. In the example in Figure 1 we consider the sentence *'Tune the PID Controller'*. The word *Tune* has an outgoing edge of type 'dobj' to the word *Controller*, implying that it is the direct object of the verb tune. The parse tree correctly identifies that the term *PID* qualifies the noun *Controller*, indicating that they must be treated as a compound word.

2.5. Neural Network Models

This section briefly highlights the main models used in this work.

2.5.1. Bidirectional LSTM

Recurrent Neural Networks (RNNs) are known to successfully process short sequences learning dependencies among the current input and previous time steps. LSTMs are special RNNs that are able to learn long-range relationships between values at the beginning and end of a sequence. In both these however, the learnt representations of a word depend only on those that come before it. In case of Bidirectional LSTMs (BiLSTMs), the model is capable of utilising contextual information from both sides of the input, producing more meaningful output.

2.5.2. Node2Vec

Node2Vec [10] is an unsupervised algorithm that provides low-dimensional vector representations of each node in a graph based on its adjacent nodes. The algorithm tries to preserve the structure in the graph in that nodes which are similar within the graph yield similar embeddings in the embedding space. In this work,

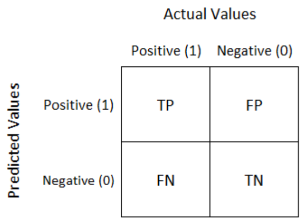

Fig. 2 Confusion Matrix

Node2Vec is used to produce embeddings for each word in a sentence based on its dependency parse graph. This is explained further in Section 3.

2.6. Metrics

Following are the metrics used to evaluate the performance of our models:

2.6.1. Accuracy

Accuracy is the number of correctly labelled points out of all data points. From Figure 2 it is defined as the number of true positives (TP) and true negatives (TN) divided by the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It is given by: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

2.6.2. Precision

Precision is a metric that tells us what proportion of positive identifications are actually correct. From Figure 2, it is defined as the number of true positives (TP) by the number of true and false positives (TP and FP). It is given by: $Precision = \frac{TP}{TP+FP}$

2.6.3. Recall

Recall is a metric that conveys the proportion of actual positives the model was able to identify correctly. From Figure 2, it is defined as the number of true positives (TP) by the number of true positives (TP) and false negatives (FN). It is given by: $Recall = \frac{TP}{TP+FN}$

2.6.4. F1-Score

The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean, balancing the trade-off between them. It is calculated as follows: It is given by: $F_1 = \frac{2(P*R)}{P+R}$, where, P = the precision and R = the recall of the classification model

3. PROPOSED METHODOLOGY

The aim of this work is to be able to classify each word as belonging to a cause phrase (C), effect phrase (E), causal connective (CC) or none (O). Earlier works in this field have already shown the importance of linguistic features, or context information, in accurate prediction of these tags. In this work we show that feeding dependency information (i.e., which entity in the sentence depends on which other entities) to the model in addition to contextual information (i.e., which words does each word occur with) helps the model in identifying more accurately which entity is the cause and which is the effect. Preference is also given to models having fewer parameters as compared to models typically used for the task.

3.1. Context Information

Context is defined as words that a given word cooccurs with. For e.g., In the sentence: *A small change in the* **set-point** *led to a large increase* in temperature, the context for the word "set-point" with a context-window of 5 consists of the italicized words. The key-insight here is that words in the context window of a given word provide additional insight into the semantics of the word, helping the model understand it better. Another factor that necessitates the use of context information is the need to locate words that come together to form the cause or effect phrase. For e.g., in the sentence above, *small change in the set-point* forms the cause phrase while *large increase in temperature* is the effect phrase. Here, the words – in, the, small and large are commonplace words occurring in many sentences and express no causality on their own. However, on coming together with other words, these become important parts of the cause or effect phrase.

3.2. Dependency Information

Dependency relations from parse trees help obtain information on how entities in a sentence relate to eachother and affect eachother. For e.g., consider the sentence *Rita hit the man with a telescope.* Here, it is important to know whether the telescope was the instrument used to hit the man or whether telescope is an identifier for the man Rita hit. The parse trees for both these interpretations have been shown below.

3.2.1. Interpretation-1

According to this dependency parse graph, telescope is the instrument with which Rita hit the man. Here, $PROPN$ = Proper noun, DET = determiner, ADP = Adposition, $dobj$ = Direct object of verb phrase, $pobj$ = Object of preposition. The tree tells us that the action of "hit" was done by Rita, with the help of a telescope, and it was done on the man. Thus, Rita is the subject, the man is the object, and the telescope is the instrument.

3.2.2. Interpretation-2

According to this dependency parse graph, Rita hit the man who had a telescope with him. The notation being the same as in Interpretation-1, this tree tells us that the action of "hit" was done by Rita, and it was done on the man to whom the telescope belonged to.

Thus, based on the above mentioned interpretations, it is clearly seen that the meaning of the sentence changes with changes in the parse tree and thus to make any conclusive remark on which entity affects which entity in the sentence, a parse tree would prove to be useful.

It is important to note that the linguistic features for both these interpretations would remain the same, it is only the dependence between words that changes the meaning. Although dependence is crucial in detecting relations and extracting causality, it cannot be solely relied

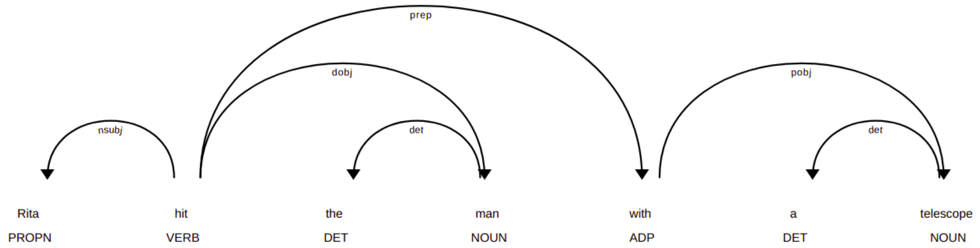

Fig. 3 Dependency Parse Tree for Interpretation-1

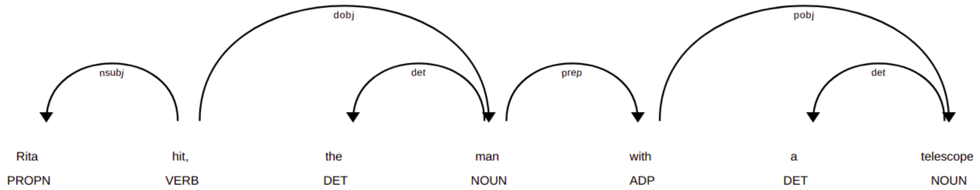

Fig. 4 Dependency Parse Tree for Interpretation-2

on since it does not maintain the sequence in which the words occur in the sentence, making it difficult to tag all words in a cause or effect phrase contiguously.

3.3. Procedure

The corpus is first cleaned to remove any noisy words or punctuations. Following this, dependency parse trees for each sentence is generated, along with linguistic features for each word. The linguistic features used include - part of speech tags, WordNet hierarchy, i.e., one of 45 fundamental classes each entity belongs to such as (substance, time, feeling, etc), IOB Code (for named entity recognition), etc. A total of 19 such linguistic features are used which are then one-hot encoded into 145 features. Further, we transform each word into its semantic representation using 50-dimensional GloVE Embeddings and each node in the dependency graph into its 20-dimensional vector representation using Node2vec. The Node2Vec model on dependency graphs with nodes being words and edges being the dependency relation between them. It generates unsupervised representations for each of these nodes which is concatenated with the word embedding and linguistic embedding for each word and then finally fed to the BiLSTM model, as can be seen in Figure 5, which uses words surrounding a given input word, along with the provided features to come up with vector representations exhibiting causal relations. These vectors are then fed into a dense layer to obtain the final output tags - C, E, CC and O for each word in the sentence.

4. RESULTS AND DISCUSSION

To demonstrate the effectiveness of the proposed architecture we perform experiments on the following datasets: SemEval 2010 task 8 [12] and BECAUSE 2.0 [13]. For each of these data sets the optimal architecture and hyperparameters differ and thus, training is performed separately for each of these. To demonstrate the effect of each of linguistic and dependence features, we show the performance of the model with linguistic and dependence features ($BiLSTM_{LD}$), with linguistic features only ($BiLSTM_L$) and dependence features only ($BiLSTM_D$). These are compared against a baseline of a BiLSTM with conditional random field model without using any linguistic or dependence features ($BiLSTM_{CRF}$).Here, accuracy is not a reliable metric since there is class imbalance among the different classes C, E, CC and O, and accuracy of incompetent models would also be high. In order to balance precision and recall, the primary metric of comparison used in this work is f_1 score.

4.1. SemEval 2010 Task 8

The corpus consists of 1003 Cause-Effect sentences. 70% of these sentences are used for training, 15% is used for validation and the remaining 15% is used for test. The training set has 3566 unique terms while the validation and test sets have 1238 and 1252 unique words each. The number of overlapping terms between the train and test set are 731. The cause/effect tags provided with the dataset annotate single words as cause and effect as opposed to phrases. This is a case of implicit causal tagging, where the causal connective is untagged even when present. Thus, the tags of interest in this dataset are C, E, O. Implementing the procedure outlined in Section 3, we construct 19 linguistic features, GloVE embeddings and 20 dimensional Node2Vec vectors are generated and fed to the model. The optimal $BiLSTM_{LD}$ model for this dataset has a BiLSTM layer of size 90 with a dropout of 0.2. Additionally, all models across all datasets are training using the categorical cross-entropy loss function with the Adam Optimizer. The results for the different models are visible in Table 3. It can clearly be seen that $BiLSTM_{LD}$ outperforms all other models in terms of accuracy, recall and f_1-score. $BiLSTM_D$, however, has a higher precision. This indicates that dependency information plays a significant role in disambiguating sen-

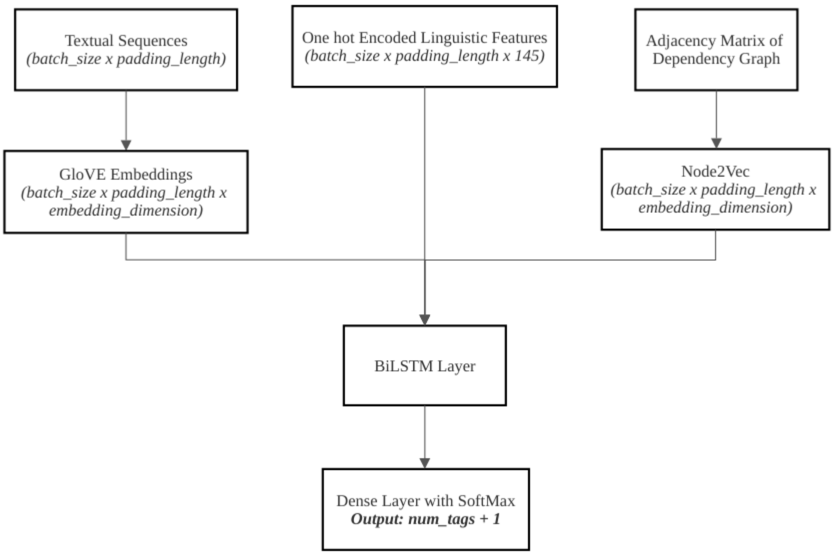

Fig. 5 Flowchart for the model: Padding Length is decided by the maximum length of sequences in a dataset; Embedding dimension for GloVE is taken as 50 while the embedding dimension for Node2Vec is taken as 20; The number of tags output by the model is 1 more due to the presence of the extra PAD character

tences of the dataset, possibly playing a more important role than context.

Examples of annotated sentences from the dataset can be seen in Table 1. In Example 1, we see that both the annotation and the ground truth match exactly while in Example 2, the model is unable to identify the cause. This may be because $excitation$ in other contexts may appear as an effect, for e.g., "Excited State", making it difficult for the model to understand this case. Further, it is seen that the model is able to detect multiple causes, as in case of Example 3, including those which have not been annotated in the ground truth. The presence of dependency information helps the model establish that both $rashes$ and $discomfort$ are similar entities in the sentence and thus, enable it to tag both even though the ground truth only labels one.

4.2. BECAUSE 2.0

The corpus consists of 1727 sentences from 5 different sources - Congressional Hearings, New York Times, Penn Tree Bank, MASC Wall Street journal. Explicit tagging is performed on this dataset with the tags of interest being C, E, O and CC. The training, validation and test split is maintained at 70%, 15%, 15% similar to the previous case. The training set has 2656 unique terms while the validation and test sets have 1007 and 908 unique words each. The number of overlapping terms between the train and test set are 524.

Implementing the procedure outlined in Section 3, we construct 19 linguistic features, GloVE embeddings and 20 dimensional Node2Vec vectors are generated and fed to the model. The optimal $BiLSTM_{LD}$ model for this dataset has a BiLSTM layer of size 32 with a dropout of 0.2.

Examples of annotated sentences from the dataset can be seen in Table 2. Similar to the previous case, in Example 1, we see that both the annotation and the ground truth match exactly while in Example 2, the model overestimates the causal connective. Finally in Example 3, it is seen that the model is able to able to detect multiple levels of causality. In the given example, the trucks entering the territory of the gunman triggered attacks. Further, these attacks stopped because of military protection. The ground truth highlights only the attacks stopping upon military intervention, but the model is also able to detect the cause of the attacks. Further, it must be noted that the model is able to detect unconventional causal connectives such as "when" and "until" which often do not appear as causal connectives unlike the more common ones such as "because" and "caused by". This suggests that the model is not learning specific word-based patterns from the training set but rather decides on a sentence-to-sentence basis, based on the meaning of the word in that context.

From Table 3 it is clearly seen that the model $BiLSTM_{LD}$ outperforms the baseline as well as other variants in terms of all the metrics.

4.3. Sample Operator Log

Finally, we demonstrate the results of the model on a portion of a sample operator log taken from the PMHSA Pipeline Hazard Dataset [14], which is from a different domain as compared to the training datasets. Due to the lack of annotated operator logs, the analysis could not be extended to multiple logs.

> *A sharp pressure pill-down resulted from the startup of a customer power plant engine. This probably led to violent lever arm movement of the regulator and the subsequent dropping of the weights. This closed the regulator valves.*

Using the operator log above we generate the GloVE and Node2Vec word embeddings along with the linguistic features. Each of these sentences are then fed to the $BiLSTM_{LD}$ model, which is chosen due to its superior

Table 1 Examples of Sentences Annotated by $BiLSTM_{LD}$ in SemEval-2010

Original Sentence	**Annotated by Model**
The $(burst)_E$ has been caused by water hammer $(pressure)_C$	The $(burst)_E$ has been caused by water hammer $(pressure)_C$
The former $(emission)_E$ arises from the primary collisional $(excitation)_C$	The former $(emission)_E$ arises from the primary collisional excitation
Sensitive pets experience $(rashes)_E$ and discomfort from ticks and $(fleas)_C$	Sensitive pets experience $(rashes)_E$ and $(discomfort)_E$ from ticks and $(fleas)_C$

Table 2 Examples of Sentences Annotated by $BiLSTM_{LD}$ in BECAUSE 2.0

Original Sentence	**Annotated by Model**
(Patrolling needs to be increased)$_E$ (because)$_{CC}$(urban crime is increasing)$_C$	(Patrolling needs to be increased)$_E$ (because)$_{CC}$ (urban crime is increasing)$_C$
(The attacks have intensified)$_E$ (since)$_{CC}$ (the govt. began the crack down on the trafficker in August)$_C$	(The attacks have intensified)$_E$ (since the)$_{CC}$ (govt. began the crack down on the trafficker in August)$_C$
(The gunman regularly attacked trucks when they entered his territory)$_E$ (until)$_{CC}$ (the military began providing security)$_C$	(The gunman regularly attacked trucks)$_E$ (when)$_{CC}$ (they entered his territory)$_C$ (until)$_C C$ (the military began providing security)$_C$

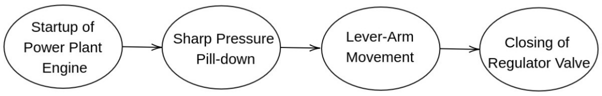

Fig. 6 Causal Graph for the Sample Operator Log

performance in terms of f_1 score, to obtain the $pressure$ and $startup$ as the effect and cause, respectively, in the first sentence; $this$ and $movement$ as cause and effect in the second sentence and finally $this$ and $regulator$ as cause and effect in the third sentence. From these, we could build a causal graph like in Figure 6, after resolving pronouns such as $this$ which bridge sentences.

Table 3 Results from all models on each of the datasets; SE: SemEval-2010; BE2.0: BECAUSE 2.0

Dataset	**Metric**	**CRF**	**L**	**D**	**LD**
SE	**Accuracy**	0.9258	0.9695	0.9704	**0.9715**
	Recall	0.6369	0.6877	0.6777	**0.7235**
	Precision	0.7441	0.7595	**0.8000**	0.7763
	F1-Score	0.6786	0.7217	0.7336	**0.7489**
BE2.0	**Accuracy**	0.5780	0.9548	0.9496	**0.9578**
	Recall	0.5780	0.7373	0.7426	**0.7755**
	Precision	0.6199	0.8079	0.7951	**0.8141**
	F1-Score	0.5740	0.7709	0.7679	**0.7942**

5. CONCLUSION

In this work we presented a novel algorithm for extracting causal information from textual data using context and dependency information. Context helps the model in understanding the sentence as a sequence of words while dependency helps relate different parts of the sentence. Collectively, they render the model more robust to different interpretations of a sentence. Needless to add, as with any complex ML / AI algorithm, the method requires sufficient quantity of good quality data. Applications on three different data sets demonstrate the potential of the proposed method. Directions for future study are along the lines of extending the proposed approach to large datasets of annotated operator logs which can then be employed for realtime causal extraction. We expect the method to be robust to unseen datasets since the grammatical structure of sentences tends to remain relatively constant across domains. Further, causal connectives also remain constant across domains helping the model gauge structure in sentences. Finally, this approach could also be extended to intersentential causal tagging provided a robust pronoun resolution model can be obtained.

REFERENCES

[1] Yang, F.; Duan, P.; Shah, S. L.; Chen, T. Capturing Connectivity and Causality in Complex Industrial Processes; Springer Science & Business Media, 2014.

[2] Kathari, S and Tangirala, A.K. (2019). Efficient Reconstruction of Granger-Causal Networks in Linear Multivariable Dynamical Processes. Industrial & Engineering Chemistry Research, 58.

[3] Sugihara, G.; May, R.; Ye, H.; Hsieh, C.-h.; Deyle, E.; Fogarty, M. and Munch, S (2012). Detecting Causality in Complex Ecosystems. Science, 338, 496-500

[4] Garcia D, EDF-DER, IMA-TIEM (2006). COATIS, an NLP system to locate expressions of actions connected by causality links, vol 1319, Springer, chap BFb0026799, pp 347–352.

[5] Khoo CSG, Chan S and Niu Y (2000). Extracting causal knowledge from a medical database using graphical patterns. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Hong Kong, pp 336–343,

[6] Lin Z, Kan MY and Ng HT (2009). Recognizing implicit discourse relations in the Penn Discourse Treebank. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, pp 343–351

[7] Zhao S, Liu T, Zhao S, Chen Y, Nie JY (2016) Event causality extraction based on connectives analysis. Neurocomputing 173:1943–1950,

[8] Dasgupta, T., Saha, R., Dey, L. and Naskar, A. (2018). Automatic extraction of causal relations from text using linguistically informed Deep Neural Networks. Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue.

[9] J. Pennington, R. Socher, and C.D. Manning (2014). GloVe: Global Vectors for Word Representation.

[10] A. Grover and J. Leskovec (2016). Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 855–864.

[11] Zhang Y, Qi P and Manning CD (2018). Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, pp 2205–2215,

[12] Hendrickx, I. (2019). SemEval-2010 Task 8: Multi-Way Classification of Semantic [Dataset]. https://arxiv.org/abs/1911.10422

[13] Dunietz, J., Levin, L.S., Carbonell, J.G. (2017). The BECauSE Corpus 2.0: Annotating Causality and Overlapping Relations. LAW@ACL.

[14] Pipeline and Hazardous Materials Safety Administration (2019). https://www.phmsa.dot.gov/data-and-statistics/pipeline/pipeline-incident-20-year-trends