Detecting and Explaining Causes From Text For a Time Series Event

Dongyeop Kang, Varun Gangal, Ang Lu, Zheng Chen, Eduard Hovy
Language Technology Institute
Carnegie Mellon University
{dongyeok,vgangal,alu1,zhengc1,hovy}@cs.cmu.edu

Abstract

Explaining underlying causes or effects about events is a challenging but valuable task. We define a novel problem of generating explanations of a time series event by (1) searching cause and effect relationships of the time series with textual data and (2) constructing a connecting chain between them to generate an explanation. To detect causal features from text, we propose a novel method based on the Granger causality of time series between features extracted from text such as N-grams, topics, sentiments, and their composition. The generation of the sequence of causal entities requires a commonsense causative knowledge base with efficient reasoning. To ensure good interpretability and appropriate lexical usage we combine symbolic and neural representations, using a neural reasoning algorithm trained on commonsense causal tuples to predict the next cause step. Our quantitative and human analysis show empirical evidence that our method successfully extracts meaningful causality relationships between time series with textual features and generates appropriate explanation between them.

1 Introduction

Producing true causal explanations requires deep understanding of the domain. This is beyond the capabilities of modern AI. However, it is possible to collect large amounts of causally related events, and, given powerful enough representational variability, to construct cause-effect chains by selecting individual pairs appropriately and linking them together. Our hypothesis is that chains composed of locally coherent pairs can suggest overall causation.

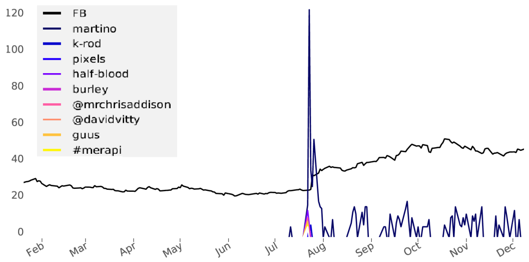


Figure 1: Example of causal features for Facebook's stock change in 2013. The causal features (e.g., *martino*, *k-rod*) rise before the Facebook's rapid stock rise in August.

In this paper, we view *causality* as (commonsense) cause-effect expressions that occur frequently in online text such as news articles or tweets. For example, "*greenhouse gases causes global warming*" is a sentence that provides an 'atomic' link that can be used in a larger chain. By connecting such causal facts in a sequence, the result can be regarded as a *causal explanation* between the two ends of the sequence (see Table 1 for examples).

This paper makes the following contributions:

- we define the problem of causal explanation generation,
- we detect causal features of a time series event (CSPIKES) using Granger (Granger, 1988) method with features extracted from text such as N-grams, topics, sentiments, and their composition,
- we produce a large graph called CGRAPH of local cause-effect units derived from text and develop a method to produce causal explanations by selecting and linking appropriate units, using neural representations to enable unit matching and chaining.

Table 1: Examples of generated causal explanation between some temporal causes and target companies' stock prices.

party $\xmapsto{cut}$ budget_cuts $\xmapsto{lower}$ budget_bill $\xmapsto{decreas}$ republicans $\xmapsto{caus}$ obama $\xmapsto{leadto}$ facebook_polls $\xmapsto{caus}$ **facebook's stock** $\downarrow$

The problem of causal explanation generation arises for systems that seek to determine causal factors for events of interest automatically. For given time series events such as companies' stock market prices, our system called CSPIKES detects events that are deemed causally related by time series analysis using Granger Causality regression (Granger, 1988). We consider a large amount of text and tweets related to each company, and produces for each company time series of values for hundreds of thousands of word n-grams, topic labels, sentiment values, etc. Figure 1 shows an example of causal features that temporally causes Facebook's stock rise in August.

However, it is difficult to understand how the statistically verified factors actually cause the changes, and whether there is a latent causal structure relating the two. This paper addresses the challenge of finding such latent causal structures, in the form of *causal explanations* that connect the given cause-effect pair. Table 1 shows example causal explanation that our system found between *party* and *Facebook's stock fall (↓)*.

To construct a general causal graph, we extract all potential causal expressions from a large corpus of text. We refer to this graph as CGRAPH. We use FrameNet (Baker et al., 1998) semantics to provide various causative expressions (verbs, relations, and patterns), which we apply to a resource of $183,253,995$ sentences of text and tweets. These expressions are considerably richer than previous rule-based patterns (Riaz and Girju, 2013; Kozareva, 2012). CGRAPH contains 5,025,636 causal edges.

Our experiment demonstrates that our causality detection algorithm outperforms other baseline methods for forecasting future time series values. Also, we tested the neural reasoner on the inference generation task using the BLEU score. Additionally, our human evaluation shows the relative effectiveness of neural reasoners in generating appropriate lexicons in explanations.

2 CSPIKES: Temporal Causality Detection from Textual Features

The objective of our model is, given a target time series y, to find the best set of textual features $F = \{f_1, ..., f_k\} \subseteq X$, that maximizes sum of causality over the features on y, where X is the set of all features. Note that each feature is itself a time series:

$$\arg\max_F \mathbf{C}(y, \Phi(X, y)) \tag{1}$$

where $\mathbf{C}(y, x)$ is a causality value function between y and x, and Φ is a linear composition function of features f. Φ needs target time series y as well because of our graph based feature selection algorithm described in the next sections.

We first introduce the basic principles of Granger causality in Section 2.1. Section 2.2 describes how to extract good source features $F = \{f_1, ..., f_k\}$ from text. Section 2.3 describes the causality function $\mathbf{C}$ and the feature composition function Φ.

2.1 Granger Causality

The essential assumption behind Granger causality is that a cause must occur before its effect, and can be used to predict the effect. Granger showed that given a target time series y (effect) and a source time series x (cause), *forecasting* future target value y_t with both past target and past source time series $E(y_t|y_{<t}, x_{<t})$ is significantly powerful than with only past target time series $E(y_t|y_{<t})$ (plain auto-regression), if x and y are indeed a cause-effect pair. First, we learn the parameters α and β to maximize the prediction expectation:

$$E(y_t|y_{<t}, x_{t-l}) = \sum_{j=1}^{m} \alpha_j y_{t-j} + \sum_{i=1}^{n} \beta_i x_{t-i} \tag{2}$$

where i and j are size of lags in the past observation. Given a pair of causes x and a target y, if β has magnitude significantly higher than zero (according to a confidence threshold), we can say that x causes y.

2.2 Feature Extraction from Text

Extracting meaningful features is a key component to detect causality. For example, to predict future trend of presidential election poll of *Donald Trump*, we need to consider his past poll data as well as people's reaction about his pledges such as *Immigration*, *Syria* etc. To extract such "good" features crawled from on-line media data, we propose three different types of features: F_{words}, F_{topic}, and F_{senti}.

F_{words} is time series of N-gram words that reflect popularity of the word over time in on-line media. For each word, the number of items (e.g., tweets, blogs and news) that contains the N-gram word is counted to get the day-by-day time series. For example, $x^{Michael\_Jordan} = [12, 51, ..]$ is a time series for a bi-gram word *Michael Jordan*. We filter out stationary words by using simple measures to estimate how dynamically the time series of each word changes over time. Some of the simple measures include Shannon entropy, mean, standard deviation, maximum slope, and number of rise and fall peaks.

F_{topic} is time series of latent topics with respect to the target time series. The latent topic is a group of semantically similar words as identified by a standard topic clustering method such as LDA (Blei et al., 2003). To obtain temporal trend of the latent topics, we choose the top ten frequent words in each topic and count their occurrence in the text to get the day-by-day time series. For example, $x^{healthcare}$ means how popular the topic *healthcare* that consists of *insurance*, *obamacare* etc, is through time.

F_{senti} is time series of sentiments (positive or negative) for each topic. The top ten frequent words in each topic are used as the keywords, and tweets, blogs and news that contain at least one of these keywords are chosen to calculate the sentiment score. The day-by-day sentiment series are then obtained by counting positive and negative words using OpinionFinder (Wilson et al., 2005), and normalized by the total number of the items that day.

2.3 Temporal Causality Detection

We define a causality function $\mathbf{C}$ for calculating causality score between target time series y and source time series x. The causality function $\mathbf{C}$ uses Granger causality (Granger, 1988) by fitting the two time series with a Vector AutoRegressive model with exogenous variables (VARX) (Hamilton, 1994): $y_t = \alpha y_{t-l} + \beta x_{t-l} + \epsilon_t$ where ϵ_t is a white Gaussian random vector at time t and l is a lag term. In our problem, the number of source time series x is not single so the prediction happens in the k multi-variate features $X = (f_1, ...f_k)$ so:

$$y_t = \alpha y_{t-l} + \boldsymbol{\beta}(f_{1,t-l} + ... + f_{k,t-l}) + \epsilon_t \quad (3)$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is the coefficient matrix of the target y and source X time series respectively, and ϵ is a residual (prediction error) for each time series. $\boldsymbol{\beta}$ means contributions of each lagged feature $f_{k,t-l}$ to the predicted value y_t. If the variance of $\boldsymbol{\beta_k}$ is reduced by the inclusion of the feature terms $f_{k,t-l} \in X$, then it is said that $f_{k,t-l}$ Granger-causes y.

Our causality function $\mathbf{C}$ is then $\mathbf{C}(y, f, l) = \Delta(\beta_{y,f,l})$ where Δ is change of variance by the feature f with lag l. The total Granger causality of target y is computed by summing the change of variance over all lags and all features:

$$\mathbf{C}(y, X) = \sum_{k,l} \mathbf{C}(y, f_k, l) \quad (4)$$

We compose best set of features Φ by choosing top k features with highest causality scores for each target y. In practice, due to large amount of computation for pairwise Granger calculation, we make a bipartite graph between features and targets, and address two practical problems: *noisiness* and *hidden edges*. We filter out noisy edges based on TFIDF and fill out missing values using non-negative matrix factorization (NMF) (Hoyer, 2004).

3 CGraph Construction

Formally, given source x and target y events that are causally related in time series, if we could find a sequence of cause-effect pairs $(x \mapsto e_1)$, $(e_1 \mapsto e_2)$, ... $(e_t \mapsto y)$, then $e_1 \mapsto e_2, ... \mapsto e_t$ might be a good causal explanation between x and y. Section 3 and 4 describe how to bridge the causal gap between given events (x, y) by (1) constructing a large general cause-effect graph (CGraph) from text, (2) linking the given events to their equivalent entities in the causal graph by finding the internal paths $(x \mapsto e_1, ...e_t \mapsto y)$ as causal explanations, using neural algorithms.

CGraph is a knowledge base graph where edges are directed and causally related between entities. To address less representational variability of rule based methods (Girju, 2003; Blanco et al., 2008; Sharp et al., 2016) in the causal graph construction, we used FrameNet (Baker et al., 1998) semantics. Using a semantic parser such

Table 2: Example (relation, cause, effect) tuples in different categories (manually labeled): *general*, *company*, *country*, and *people*. FrameNet labels related to causation are listed inside parentheses. The number of distinct relation types are 892.

	Relation	**Cause ↦ Effect**	
General	causes (Causation)	the virus (Cause)	aids (Effect)
	cause (Causation)	greenhouse gases (Cause)	global warming (Effect)
	forced (Causation)	the reality of world war ii (Cause)	the cancellation of the olympics (Effect)
Company	heats (Cause_temperature_change)	microsoft vague on windows (Item)	legislation battle (Agent)
	promotes (Cause_change_of_position_on_a_scale)	chrome (Item)	google (Agent)
	makes (Causation)	twitter (Cause)	love people you 've never met facebook (Effect)
Country	developing (Cause_to_make_progress)	north korea (Agent)	nuclear weapons (Project)
	improve (Cause_to_make_progress)	china (Agent)	its human rights record (Project)
	forced (Causation)	war with china (Cause)	the japanese to admit , in july 1938 (Effect)
People	attracts (Cause_motion)	obama (Agent)	more educated voters (Theme)
	draws (Cause_motion)	on america 's economic brains (Goal)	barack obama (Theme)
	made (Causation)	michael jordan (Cause)	about $ 33 million (Effect)

as SEMAFOR (Chen et al., 2010) that produces a FrameNet style analysis of semantic predicate-argument structures, we could obtain lexical tuples of causation in the sentence. Since our goal is to collect only causal relations, we extract total 36 causation related frames[1] from the parsed sentences.

Table 3: Number of sentences parsed, number of entities and tuples, and number of edges (*KB-KB*, *KBcross*) expanded by Freebase in CGRAPH.

# Sentences	# Entities	# Tuples	# *KB-KB*	# *KBcross*
183,253,995	5,623,924	5,025,636	470,250	151,752

To generate meaningful explanations, high coverage of the knowledge is necessary. We collect six years of tweets and NYT news articles from 1989 to 2007 (See Experiment section for details). In total, our corpus has 1.5 billion tweets and 11 million sentences from news articles. The Table 3 has the number of sentences processed and number of entities, relations, and tuples in the final CGRAPH.

Since the tuples extracted from text are very noisy [2], we constructed a large causal graph by linking the tuples with string match and filter out the noisy nodes and edges based on some graph statistics. We filter out nodes with very high degree that are mostly stop-words or auto-generated sentences. Too long or short sentences are also filtered out. Table 2 shows the (case, relation, effect) tuples with manually annotated categories such as *General*, *Company*, *Country*, and *People*.

4 Causal Reasoning

To generate a causal explanation using CGRAPH, we need traversing the graph for finding the path between given source and target events. This section describes how to efficiently traverse the graph by expanding entities with external knowledge base and how to find (or generate) appropriate causal paths to suggest an explanation using symbolic and neural reasoning algorithms.

4.1 Entity Expansion with Knowledge Base

A simple choice for traversing a graph are the traditional graph searching algorithms such as Breadth-First Search (BFS). However, the graph searching procedure is likely to be incomplete (*low recall*), because simple string match is insufficient to match an effect to all its related entities, as it misses out in the case where an entity is semantically related but has a lexically different name.

To address the *low recall* problem and generate better explanations, we propose the use of knowledge base to augment our text-based causal graph with real-world semantic knowledge. We use Freebase (Google, 2016) as the external knowledge base for this purpose. Among 1.9 billion edges in original Freebase dump, we collect its first and second hop neighbours for each target events.

While our CGRAPH is lexical in nature, Freebase entities appear as identifiers (MIDs). For entity linking between two knowledge graphs, we need to annotate Freebase entities with their lexical names by looking at the wiki URLs. We refer to the edges with freebase expansion as *KB-KB* edges, and link the *KB-KB* with our CGRAPH us-

[1] Causation, Cause_change, Causation_scenario, Cause_benefit_or_detriment, Cause_bodily_experience, etc.

[2] SEMAFOR has around 62% of accuracy on held-out set.

ing lexical matching, referring as *KBcross* edges (See Table 3 for the number of the edges).

4.2 Symbolic Reasoning

Simple traversal algorithms such as BFS are infeasible for traversing the CGraph due to the large number of nodes and edges. To reduce the search space k in $e_t \mapsto \{e^1_{t+1}, ...e^k_{t+1}\}$, we restricted our search by depth of paths, length of words in entity's name, and edge weight.

Algorithm 1 Backward Causal Inference. y is target event, d is depth of BFS, l is lag size, BFS_{back} is Breadth-First search for one depth in backward direction, and $\sum_l \mathbf{C}$ is sum of Granger causality over the lags.

```
$\mathbb{S} \leftarrow y, d = 0$
while ($\mathbb{S} = \varnothing$) or ($d > D_{max}$) do
    $\{e^1_{-d}, ...e^k_{-d}\} \leftarrow BFS_{back}(\mathbb{S})$
    $d = d + 1, \mathbb{S} \leftarrow \varnothing$
    for $j$ in $\{1, ..., k\}$ do
        if $\sum_l \mathbf{C}(y, e^j_{-d}, l) < \epsilon$ then $\mathbb{S} \leftarrow e^j_{-d}$
```

For more efficient inference, we propose a backward algorithm that searches potential causes (instead of effects) $\{e^1_t, ...e^k_t\} \leftarrow\!\shortmid\, e_{t+1}$ starting from the target node $y = e_{t+1}$ using Breadth-first search (BFS). It keeps searching backward until the node e^j_i has less Granger confident causality with the target node y (See Algorithm 4 for causality calculation). This is only possible because our system has temporal causality measure between two time series events. See Algorithm 1 for detail.

4.3 Neural Reasoning

While symbolic inference is fast and straightforward, the sparsity of edges may make our inference semantically poor. To address the *lexical sparseness*, we propose a lexically relaxed reasoning using a neural network.

Inspired by recent success on alignment task such as machine translation (Bahdanau et al., 2014), our model learns the causal alignment between cause phrase and effect phrase for each type of relation between them. Rather than traversing the CGraph, our neural reasoner uses CGraph as a training resource. The encoder, a recurrent neural network such as LSTM (Hochreiter and Schmidhuber, 1997), takes the causal phrase while the decoder, another LSTM, takes the effectual phrase with their relation specific attention.

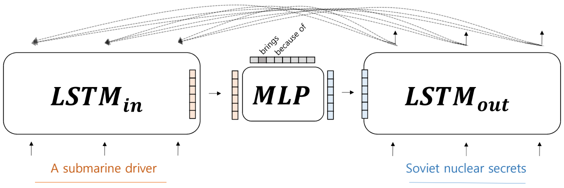


Figure 2: Our neural reasoner. The encoder takes causal phrases and decoder takes effect phrases by learning the causal alignment between them. The MLP layer in the middle takes different types of FrameNet relation and locally attend the cause to the effect w.r.t the relation (e.g., "because of", "led to", etc).

In original attention model (Bahdanau et al., 2014), the contextual vector c is computed by $c_i = a_{ij} * h_j$ where h_j is hidden state of causal sequence at time j and a_{ij} is soft attention weight, trained by feed forward network $a_{ij} = FF(h_j, s_{i-1})$ between input hidden state h_j and output hidden state s_{i-1}. The global attention matrix a, however, is easy to mix up all local alignment patterns of each relation.

For example, a tuple, *(north korea (Agent)* $\xmapsto[(Cause\_to\_make\_progress)]{developing}$ *nuclear weapons (Project))*, is different with another tuple, *(chrome (Item)* $\xmapsto[(Cause\_change\_of\_position)]{promotes}$ *google (Agent))* in terms of local type of causality. To deal with the *local attention*, we decomposed the attention weight a_{ij} by relation specific transformation in feed forward network:

$$a_{ij} = FF(h_j, s_{i-1}, r)$$

where FF has relation specific hidden layer and $r \in R$ is a type of relation in the distinct set of relations R in training corpus (See Figure 2).

Since training only with our causal graph may not be rich enough for dealing various lexical variation in text, we use pre-trained word embedding such as word2vec (Mikolov and Dean, 2013) trained on GoogleNews corpus[3] for initialization. For example, given a cause phrase *weapon equipped*, our model could generate multiple effect phrases with their likelihood: *(*$\xmapsto[0.54]{result}$*war)*, *(*$\xmapsto[0.12]{force}$*army reorganized)*, etc, even though there are no tuples exactly matched in CGraph.

[3] https://code.google.com/archive/p/word2vec/

Table 4: Examples of F_{words} with their temporal dynamics: Shannon entropy, mean, standard deviation, slope of peak, and number of peaks.

	entropy	mean	STD	max_slope	#-peaks
#lukewilliamss	0.72	22.01	18.12	6.12	31
happy_thanksgiving	0.40	61.24	945.95	3423.75	414
michael_jackson	0.46	141.93	701.97	389.19	585

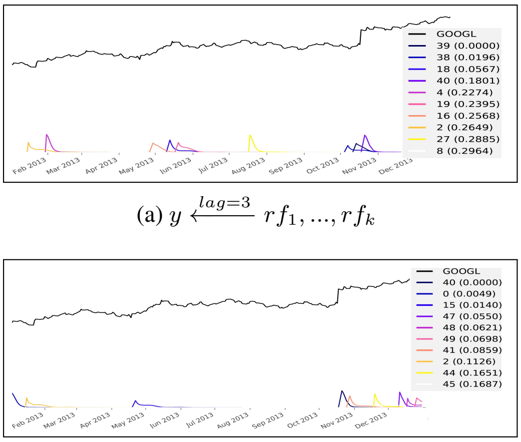
We trained our neural reasoner in either forward or backward direction. In prediction, decoder inferences by predicting effect (or cause) phrase in forward (or backward) direction. As described in the Algorithm 1, the backward inference continue predicting the previous causal phrases until it has high enough Granger confidence with the target event.

5 Experiment

Data. We collect on-line social media from tweets, news articles, and blogs. Our Twitter data has one million tweets per day from 2008 to 2013 that are crawled using Twitter's Garden Hose API. News and Blog dataset have been crawled from 2010 to 2013 using Google's news API. For target time series, we collect companies' stock prices in NASDAQ and NYSE from 2001 until present for 6,200 companies. For presidential election polls, we collect polling data of the 2012 presidential election from 6 different websites, including USA Today , Huffington Post, Reuters, etc.

Features. For N-gram word features F_{word},we choose the spiking words based on their temporal dynamics (See Table 4). For example, if a word is too frequent or the time series is too burst, the word should be filtered out because the trend is too general to be an event. We choose five types of temporal dynamics: Shannon entropy, mean, standard deviation, maximum slope of peak, and number of peaks; and delete words that have too low or high entropy, too low mean and deviation, or the number of peaks and its slope is less than a certain threshold. Also, we filter out words whose frequency is less than five. From the $1,677,583$ original words, we retain $21,120$ words as final candidates for F_{words} including uni-gram and bi-gram words.

For sentiment F_{senti} and topic F_{topic} features, we choose 50 topics generated for both politicians and companies separately using LDA, and then use top 10 words for each topic to calculate sentiment score for this topic. Then we can analyze the causality between sentiment series of a specific topic and collected time series.



(a) $y \xleftarrow{lag=3} rf_1, ..., rf_k$

(b) $y \xrightarrow{lag=3} rf_1, ..., rf_k$

Figure 3: Random causality analysis on **Googles**'s stock price change (y) and randomly generated features (rf) during 2013-01-01 to 2013-12-31. (a) shows how the random features rf cause the target y, while (b) shows how the target y causes the random features rf with lag size of 3 days. The color changes according to causality confidence to the target (blue is the strongest, and yellow is the weakest). The target time series has y scale of prices, while random features have y scale of causality degree $\mathbf{C}(y, rf) \subset [0, 1]$.

Tasks. To show validity of causality detector, first we conduct random analysis between target time series and randomly generated time series. Then, we tested forecasting stock prices and election poll values with or without the detected textual features to check effectiveness of our causal features. We evaluate our reasoning algorithm for generation ability compared to held-out cause-effect tuples using BLEU metric. Then, for some companies' time series, we describe some qualitative result of some interesting causal text features found with Granger causation and explanations generated by our reasoners between the target and the causal features. We also conducted human evaluation on the explanations.

5.1 Random Causality Analysis

To check whether our causality scoring function $\mathbf{C}$ detects the temporal causality well, we conduct a random analysis between target time series and randomly generated time series (See Figure 3). For Google's stock time series, we regularly move window size of 30 over the time and generate five days of time series with a random peak strength using a SpikeM model (Matsubara et al., 2012)[4]. The color of random time series rf changes from blue to yellow according to causality degree with the target $\mathbf{C}(y, rf)$. For example, blue is the strongest causality with target time series, while yellow is the weakest.

We observe that the strong causal (blue) features are detected just before (or after) the rapid rise of Google' stock price on middle October in (a) (or in (b)). With the lag size of three days, we observe that the strength of the random time series gradually decreases as it grows apart from the peak of target event. The random analysis shows that our causality function $\mathbf{C}$ appropriately finds cause or effect relation between two time series in regard of their strength and distance.

5.2 Forecasting with Textual Features

Table 5: Forecasting errors (RMSE) on **Stock** and **Poll** data with time series only (*SpikeM* and *LSTM*) and with time series plus text feature (*random*, *words*, *topics*, *sentiment*, and *composition*).

		Time Series		Time Series + Text				
	Step	SpikeM	LSTM	$\mathbf{C}_{rand}$	$\mathbf{C}_{words}$	$\mathbf{C}_{topics}$	$\mathbf{C}_{senti}$	$\mathbf{C}_{comp}$
Stock	1	102.13	6.80	3.63	2.97	3.01	3.34	1.96
	3	99.8	7.51	4.47	4.22	4.65	4.87	3.78
	5	97.99	7.79	5.32	5.25	5.44	5.95	5.28
Poll	1	10.13	1.46	1.52	1.27	1.59	2.09	1.11
	3	10.63	1.89	1.84	1.56	1.88	1.94	1.49
	5	11.13	2.04	2.15	1.84	1.88	1.96	1.82

We use time series forecasting task as an evaluation metric of whether our textual features are appropriately causing the target time series or not. Our feature composition function Φ is used to extract good causal features for forecasting. We test forecasting on stock price of companies (**Stock**) and predicting poll value for presidential election (**Poll**). For stock data, We collect daily closing stock prices during 2013 for ten IT companies[5]. For poll data, we choose ten candidate politicians [6] in the period of presidential election in 2012.

For each of stock and poll data, the future trend of target is predicted only with target's past time series or with target's past time series and past time series of textual features found by our system. Forecasting only with target's past time series uses *SpikeM* (Matsubara et al., 2012) that models a time series with small number of parameters and simple *LSTM* (Hochreiter and Schmidhuber, 1997; nne, 2015) based time series model. Forecasting with target and textual features' time series use Vector AutoRegressive model with exogenous variables (VARX) (Hamilton, 1994) from different composition function such as $\mathbf{C}_{random}$, $\mathbf{C}_{words}$, $\mathbf{C}_{topics}$, $\mathbf{C}_{senti}$, and $\mathbf{C}_{composition}$. Each composition function except $\mathbf{C}_{random}$ uses top ten textual features that causes each target time series. We also tested LSTM with past time series and textual features but VARX outperforms LSTM.

Table 6: Beam search results in neural reasoning. These examples could be filtered out by graph heuristics before generating final explanation though.

Cause↦Effect in CGRAPH	Beam Predictions
the dollar's $\xmapsto{caus}$ against the yen	[1]$\xmapsto{caus}$ against the yen [2]$\xmapsto{caus}$ against the dollar [3]$\xmapsto{caus}$ against other currencies
without any exercise $\xmapsto{caus}$ news article	[1]$\xmapsto{leadto}$ a difference [2]$\xmapsto{caus}$ the risk [3]$\xmapsto{make}$ their weight

Table 5 shows root mean square error (RMSE) for forecasting with different step size (time steps to predict), different set of features, and different regression algorithms on stock and poll data. The forecasting error is summation of errors over moving a window (30 days) by 10 days over the period. Our $\mathbf{C}_{composition}$ method outperforms other time series only models and time series plus text models in both stock and poll data.

5.3 Generating Causality with Neural Reasoner

The reasoner needs to predict the next effect phrase (or previous cause phrase) so the model should be evaluated in terms of generation task. We used the BLEU (Papineni et al., 2002) metric to evaluate the predicted phrases on held out phrases in our CGRAPH . Since our CGRAPH has many edges, there may be many good paths (explanations), possibly making our prediction diverse. To evaluate such diversity in prediction, we used ranking-based BLEU method on the k set of

[4] SpikeM has specific parameters for modeling a time series such as peak strength, length, etc.

[5] Company symbols used: TSLA, MSFT, GOOGL, YHOO, FB, IBM, ORCL, AMZN, AAPL and HPO

[6] Name of politicians used: Santorum, Romney, Pual, Perry, Obama, Huntsman, Gingrich, Cain, Bachmann

Table 7: BLEU ranking. Additional word representation **+WE** and relation specific alignment **+REL** help the model learn the cause and effect generation task especially for diverse patterns.

	B@1	B@3A	B@5A
S2S	10.15	8.80	8.69
S2S + WE	11.86	10.78	10.04
S2S + WE + REL	12.42	12.28	11.53

predicted phrases by beam search. For example, $B@k$ means BLEU scores for generating k number of sentences and $B@kA$ means the average of them.

Table 6 shows some examples of our beam search results when $k = 3$. Given a cause phrase, the neural reasoner sometime predicts semantically similar phrases (e.g., *against the yen*, *against the dollar*), while it sometimes predicts very diverse phrases (e.g., *a different*, *the risk*).

Table 7 shows BLEU ranking results with different reasoning algorithms: **S2S** is a sequence to sequence learning trained on CGRAPH by default, **S2S+WE** adds word embedding initialization, and **S2S+REL+WE** adds relation specific attention. Initializing with pre-trained word embeddings (**+WE**) helps us improve on prediction. Our relation specific attention model outperforms the others, indicating that different type of relations have different alignment patterns.

5.4 Generating Explanation by Connecting

Evaluating whether a sequence of phrases is reasonable as an explanation is very challenging task. Unfortunately, due to lack of quantitative evaluation measures for the task, we conduct a human annotation experiment.

Table 8 shows example causal chains for the rise (↑) and fall (↓) of companies' stock price, continuously produced by two reasoners: *SYBM* is symbolic reasoner and *NEUR* is neural reasoner.

We also conduct a human assessment on the explanation chains produced by the two reasoners, asking people to choose more convincing explanation chains for each feature-target pair. Table 9 shows their relative preferences.

6 Related Work

Prior works on causality detection (Acharya, 2014; Anand, 2014; Qiu et al., 2012) in time series data (e.g., gene sequence, stock prices, temperature) mainly use Granger (Granger, 1988) ability for predicting future values of a time series using past values of its own and another time series. (Hlaváčková-Schindler et al., 2007) studies more theoretical investigation for measuring causal influence in multivariate time series based on the entropy and mutual information estimation. However, none of them attempts generating explanation on the temporal causality.

Previous works on text causality detection use syntactic patterns such as $X \xmapsto{verb} Y$, where the $verb$ is causative (Girju, 2003; Riaz and Girju, 2013; Kozareva, 2012; Do et al., 2011) with additional features (Blanco et al., 2008). (Kozareva, 2012) extracted cause-effect relations, where the pattern for bootstrapping has a form of $X^* \xmapsto[Z^*]{verb} Y$ from which terms X^* and Z^* was learned. The syntax based approaches, however, are not robust to semantic variation.

As a part of SemEval (Girju et al., 2007), (Mirza and Tonelli, 2016) also uses syntactic causative patterns (Mirza and Tonelli, 2014) and supervised classifier to achieve the state-of-the-art performance. Extracting the cause-effect tuples with such syntactic features or temporality (Bethard et al., 2008) would be our next step for better causal graph construction.

(Grivaz, 2010) conducts very insightful annotation study of what features are used in human reasoning on causation. Beyond the linguistic tests and causal chains for explaining causality in our work, other features such as counterfactuality, temporal order, and ontological asymmetry remain as our future direction to study.

Textual entailment also seeks a directional relation between two given text fragments (Dagan et al., 2006). Recently, (Rocktäschel et al., 2015) developed an attention-based neural network method, trained on large annotated pairs of textual entailment, for classifying the types of relations with decomposable attention (Parikh et al., 2016) or sequential tree structure (Chen et al., 2016). However, the dataset (Bowman et al., 2015) used for training entailment deals with just three categories, *contradiction*, *neutral*, and *entailment*, and focuses on relatively simple lexical and syntactic transformations (Kolesnyk et al., 2016). Our causal explanation generation task is also similar to *future scenario generation* (Hashimoto et al., 2014, 2015). Their scoring

Table 8: Example causal chains for explaining the rise (↑) and fall (↓) of companies' stock price. The temporally causal *feature* and *target* are linked through a sequence of predicted cause-effect tuples by different reasoning algorithms: a symbolic graph traverse algorithm *SYMB* and a neural causality reasoning model *NEUR*.

SYMB	**medals** $\xmapsto{match}$ gold_and_silver_medals $\xmapsto{swept}$ korea $\xmapsto{improving}$ relations $\xmapsto{widened}$ gap $\xmapsto{widens}$ **facebook** ↑
	excess $\xmapsto{match}$ excess_materialism $\xmapsto{cause}$ people_make_films $\xmapsto{make}$ money $\xmapsto{changed}$ twitter $\xmapsto{turned}$ **facebook** ↓
	clinton $\xmapsto{match}$ president_clinton $\xmapsto{raised}$ antitrust_case $\xmapsto{match}$ government's_antitrust_case_against_microsoft $\xmapsto{match}$ microsoft $\xmapsto{beats}$ **apple** ↓
NEUR	**google** $\xmapsto{forc}$ microsoft_to_buy_computer_company_dell_announces_recall_of_batteries $\xmapsto{cause}$ **microsoft** ↑
	the_deal $\xmapsto{make}$ money $\xmapsto{rais}$ at_warner_music_and_google_with_protest_videos_things $\xmapsto{caus}$ **google** ↓
	party $\xmapsto{cut}$ budget_cuts $\xmapsto{lower}$ budget_bill $\xmapsto{decreas}$ republicans $\xmapsto{caus}$ obama $\xmapsto{leadto}$ facebook_polls $\xmapsto{caus}$ **facebook** ↓
	company $\xmapsto{forc}$ to_stock_price $\xmapsto{leadto}$ investors $\xmapsto{increas}$ oracle_s_stock $\xmapsto{increas}$ **oracle** ↑

Table 9: Human evaluation on explanation chains generated by symbolic and neural reasoners.

Reasoners	SYMB	NEUR
Accuracy (%)	42.5	57.5

function uses heuristic filters and is not robust to lexical variation.

7 Conclusion

This paper defines the novel task of detecting and explaining causes from text for a time series. First, we detect causal features from online text. Then, we construct a large cause-effect graph using FrameNet semantics. By training our relation specific neural network on paths from this graph, our model generates causality with richer lexical variation. We could produce a chain of cause and effect pairs as an explanation which shows some appropriateness. Incorporating aspects such as time, location and other event properties remains a point for future work. In our following work, we collect a sequence of causal chains verified by domain experts for more solid evaluation of generating explanations.

References

2015. Neural network architecture for time series forecasting. https://github.com/hawk31/nnet-ts.

Saurav Acharya. 2014. Causal modeling and prediction over event streams.

Surya Pratap Singh Tanwar Anand, Mehndiratta. 2014. Web Metric Summarization using Causal Relationship Graph.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Steven Bethard, William J Corvey, Sara Klingenstein, and James H Martin. 2008. Building a corpus of temporal-causal structure. In *LREC*.

Eduardo Blanco, Nuria Castell, and Dan I Moldovan. 2008. Causal relation extraction. In *LREC*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Desai Chen, Nathan Schneider, Dipanjan Das, and Noah A Smith. 2010. Semafor: Frame argument resolution with log-linear models. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 264–267. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pages 177–190. Springer.

Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*, pages 76–83. Association for Computational Linguistics.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18. Association for Computational Linguistics.

Google. 2016. Freebase Data Dumps. `https://developers.google.com/freebase/data`.

Clive WJ Granger. 1988. Some recent development in a concept of causality. *Journal of econometrics*, 39(1):199–211.

Cécile Grivaz. 2010. Human judgements on causation in french texts. In *LREC*.

James Douglas Hamilton. 1994. *Time series analysis*, volume 2. Princeton university press Princeton.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, and Jong-Hoon Oh. 2015. Generating event causality hypotheses through semantic relations. In *AAAI*, pages 2396–2403.

Chikara Hashimoto, Kentaro Torisawa, Julien Kloetzer, Motoki Sano, István Varga, Jong-Hoon Oh, and Yutaka Kidawara. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *ACL (1)*, pages 987–997.

Katerina Hlaváčková-Schindler, Milan Paluš, Martin Vejmelka, and Joydeep Bhattacharya. 2007. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Patrik O Hoyer. 2004. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469.

Vladyslav Kolesnyk, Tim Rocktäschel, and Sebastian Riedel. 2016. Generating natural language inference chains. *arXiv preprint arXiv:1606.01404*.

Zornitsa Kozareva. 2012. Cause-effect relation learning. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, pages 39–43. Association for Computational Linguistics.

Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. 2012. Rise and fall patterns of information diffusion: model and implications. In *KDD*, pages 6–14.

T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

Paramita Mirza and Sara Tonelli. 2014. An analysis of causality between events and its relation to temporal information. In *COLING*, pages 2097–2106.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *The 26th International Conference on Computational Linguistics*, pages 64–75.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Huida Qiu, Yan Liu, Niranjan A Subrahmanya, and Weichang Li. 2012. Granger causality for time-series anomaly detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1074–1079. IEEE.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *Proceedings of the annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*. Citeseer.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. *arXiv preprint arXiv:1609.08097*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.