CSpikes: Detecting Causality between Time Series and Textual Data

Anonymous ACL submission

Abstract

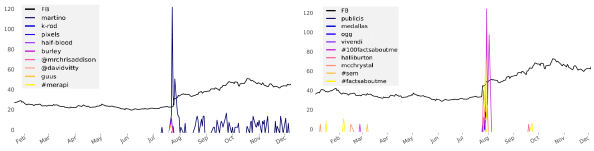
Finding cause and effect relationships between events is a very challenging problem. We search for causal relationships between time series and textual data. In this paper, we propose a novel method CSpikes based on Granger causality of time series between features extracted from text such as N-grams, topics, sentiments, and their composition. We test our algorithm on random analysis and forecasting. Both of quantitative and qualitative analysis show empirical evidence that our *Temporal-Causality graph* (TC-Graph) successfully extracts meaningful causality relationships between time series with textual features.

1 Introduction

Finding causality is an important problem in many areas, including physics, econometrics, medicine, and climate science. The most interesting application of this ability is *forecasting*. Predicting future trends based on past data is the core of stock price prediction (Granger, 1992), election poll prediction (O'Connor et al., 2010), EEG signal analysis (Kamiński et al., 2001) and many other tasks.

Our survey of the literature on causality detection in either time series data or textual data shows their differences in representation, sources, and space, but also shows underlying similarities that suggest the potential of linking the two kinds of data. The semantic complementarity of the two kinds of information offers the potential for possibly significant enhancement. The objective of this paper is to explore this potential.

In this paper we propose CSpikes, an efficient causality detection algorithm linking two time series, one called *target* with numerical information and the other called *feature* derived from text, using Granger causality (Granger, 1988). The textual *features* are extracted from news, social media, and other texts, and include word N-grams, topic labels, and sentiment values. The *target* is a sequence of numerical values (e.g., stock prices, election poll data) of which our system tries to find its cause or effect from the textual features. For example, Figure 1 shows top causal and effectual *features* derived from N-gram words for a target company *Facebook*'s stock change during 2013.



(a) Facebook $\xleftarrow{lag=3}$ $f_1, ..., f_k$ **(b)** $f_1, ..., f_k$ $\xleftarrow{lag=3}$ **Facebook**

Figure 1: Example of (a) causal and (b) effectual features for Facebook's stock change in 2013 with lag size of 3. The causal features (e.g., *martino*, *k-rod*, *john_henry*) rise around 3 days before the Facebook's rapid stock rise in August, while the effect features (e.g., *publicis*, *medallas*, *vivendi*) rise around 3 days after it.

To deal with the data sparsity, We propose a sparse bipartite representation called Temporal-Causality graph (TC-Graph) between the targets and features. We use TFIDF scheme for filtering out the noise features and non-negative matrix factorization for filling out hidden edges. To find the best set of features for forecasting over the TC-Graph, then we compose top features from the factorized matrices for each target.

To validate our algorithm, we conduct different tasks: random analysis, forecasting, and qualitative analysis on the detected causalities. The random analysis shows that our Granger analysis detects the causal relation between the numerical time series and randomly generated time se-

ries features. In forecasting, the combination of our textual features and numerical time-series data shows comparable accuracy with the baselines. Our qualitative analysis shows an empirical evidence that TC-GRAPH efficiently detects Granger causes and effects for both stock prices and political election polls.

In Section 2, we survey literature on temporal and linguistic causality detection and their comparison. In Section 3, we propose our methods on feature extraction, causality detection and feature composition. Finally, Section 4 shows empirical observations from random analysis, forecasting and qualitative analysis, and Section 5 concludes our paper.

2 Preliminary Study

We survey the past literature on causality detection in two fields: causality in *time series data* and *textual data*.

Temporal Causality is used in either time series or sequential data (e.g., gene sequence, stock prices, temperature). Granger (Granger, 1988) measures the ability of predicting future values of a time series using past values of its own and another time series. The causal relation between such two time series is called temporal causality. (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.

Basic assumption behind Granger causality is that cause must occur before the effect, and it can be used to predict the effect. Granger shows 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 $E(y_t|y_{<t})$. Then, the goal is to find the best parameters α and β to maximize the prediction expectation:

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

where i and j are size of lags in the past observation. Given a pair of causes x and a target y, if the estimates of the coefficients α and β are higher enough than confidence threshold, we can say that x has a causation to y in time series.

The possible applications of causality detection include constructing causal relationships for web summarization (Dhruv Anand, 2014), event mining (Acharya, 2014), and anomaly detection (Qiu et al., 2012). To the best of our knowledge, our work is the first attempt that uses Granger causality for detecting causality with textual features.

Linguistic Causality is causation found from a sequence of sentences or phrases. For example, *greehouse gases* is a cause of *global warming*. However, the conditional statement such as "if then" is not causation because the statement does not contain the antecedent to precede or coincide with the consequent in time.

Most of the early works (Girju, 2003; Riaz and Girju, 2013; Kozareva, 2012; Do et al., 2011; Blanco et al., 2008) on linguistic causality detection are based on syntactic patterns between textual fragments such as $x \overset{verb}{\longmapsto} y$ where the *verb* is causative verbs (e.g., cause, infer). (Girju, 2003) automatically discovered lexical and semantic constraints for disambiguation of causal relations in question answering system. They define three types of causative verbs: simple (e.g., cause, lead to), resultative (e.g., kill, melt), and instrumental (e.g., poison, hang). (Kozareva, 2012) extracts cause-effect relations, where the pattern for bootstrapping has a form of $x^* \overset{verb}{\underset{Z^*}{\longmapsto}} y$ from which terms X^* and Z^* can be learned. For discourse text, (Do et al., 2011) observes some discourse relations that provides additional information for causalities between events.

More broadly, linguistic causality is also called *textual entailment* which is a directional relation between text fragments (Dagan et al., 2006). Most recently, (Rocktäschel et al., 2015; Parikh et al., 2016; Chen et al., 2016) develop neural network method for recognizing entailment relations from text. However, none of the prior works bridge the textual fragments with time series data using Granger causality.

Some recent works (Schwarz and Mattern, 1994; Mirza, 2014; Mirza and Tonelli, 2014) model the causative statement with temporal constraints. However, the models cannot handle our problem where the causes are not given (hidden) so detecting them from external resource is necessary.

In summary, Table 1 shows major differences between temporal and linguistic causality in the past literature survey. Temporal causation is represented in a sequence of numerical time series data, while linguistic causation is written as a sequence of text fragments such as phrase. In causality with

Table 1: Differences between temporal causation and linguistic causality in three different aspects: *representation* of the data, relationship between *source-target*, and *space* of data representation.

Aspects	Temporal	Linguistic
Representation	time series	text
Source-Target	hidden	explicit
Space	continuous	discrete

time series, we don't know types of the causality between target and source time series, while causal statement written in text explicitly encodes the relation type such as "because of" or "force to". Lastly, time series data changes in the continuous space of time, whereas textual data is a sequence of discrete words that is often represented as bag of words by its frequency. The different properties between time series and text data give a intuition for combining the two kinds of data for complementing the semantic information.

3 CSpikes: Temporal Causality Detection from Textual Features

The objective of our model is given target time series y to find the best set of time series of textual features $X = \{f_1, ..., f_k\}$ that maximizes:

$$\arg\max_X \mathbf{C}(y, \Phi(X = \{f_1, ..., f_k\}, y)) \quad (2)$$

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

We first introduce a pipeline of our work in Section 3.1. The following Section 3.2 describes how to extract good source features $X = \{f_1, ..., f_k\}$. Then, we describe the causality function $\mathbf{C}$ in Section 3.3 and the feature composition function Φ in Section 3.4.

3.1 Pipeline

We introduce a basic pipeline of our work (See Figure 2). The *Crawler* collects target time series such as political election polls and stock prices, and textual data from on-line social media such as blogs, tweets, and news articles. Then, our NLP *Parser* parses sentences using dependency parser, named entity recognizer, POS tagger and coreference resolution. From the parsed sentences, *Feature Extractor* extracts meaningful features ($x = f_1, ...f_k$) with N-gram frequencies, sentiment values, and topic labels. The *Serieser* generates time series of each feature based on their timestamps. At last, *Regressior* finds best compositional feature set $\{f_1, ..., f_k\}$ based on composition function Φ. A recent work (Kang et al., 2017) introduces a method of generating explanation about the gap between the target and feature event using neural reasoning algorithm.

3.2 Feature Extraction from Text

Extracting meaningful features is a key component to find the temporal causalities. 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* and so on. following the Equation 1, his future poll trend y_t^{Trump} could be predicted as follows:

$$\hat{y}_t^{Trump} = \alpha y_{t-l}^{Trump} + (\beta^{Syria} x_{t-l}^{Syria} + \beta^{Immig.} x_{t-l}^{Immig.}) \quad (3)$$

where α and β are coefficients for y and x, respectively. 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[1] words. 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, $x^{Michael\_Jordan} = [0, 1, 51, 622, ..., 0]$ is a time series for a bi-gram word *Michael Jordan*. The N-gram often reflects popularity of the word over time in on-line media.

Due to the huge number of N-grams, however, taking into account all N-gram words is computationally impossible. We filter out meaningless and stationary words based on how dynamically time series of each word changes over time called *temporal dynamics*. For example, if a word has no variance in time series or only too many number of rise or fall happens, it is difficult to say that the word could be a strong cause on target time series. We used several metrics for the temporal dynamics: Shannon entropy, mean, standard deviation, maximum slope, and number of rise and fall peaks. Please see some examples in Table 2.

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 by a topic clustering method such as LDA (Blei et al., 2003). To obtain temporal trend of the latent topics, we choose

[1] In computational linguistics, an n-gram means a contiguous sequence of n items from a given sequence of text

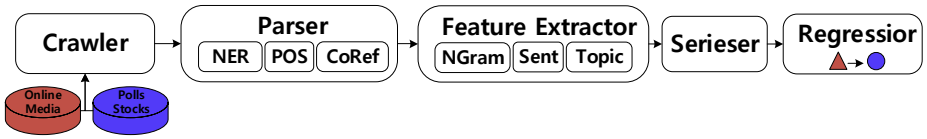


Figure 2: Pipeline of our work. The *crawler* collects social media data (e.g., twitter, blogs, news) for features and time series data (e.g., presidential election polls, companies' stock prices) for targets. The *Parser* parses the textual sentences with syntactic parser, named entity recognizer, POS tagger and coreference analyzer. The *Feature Extractor* and *Serieser* extract features from time-stamped parsed sentences and generate a time series for each feature and for each dataset. The *Regressor* then finds causal relationships between target time series and feature time series, where the features are collected through feature extraction.

the top ten frequent words in each topic and count their occurrence in the text to get the day-by-day time series. The temporal trend of each topic reflects popularity of the topics over time. 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. Finally, we can generate time series $x^{healthcare,Positive}$ of whether people from on-line media think about *healthcare* positively or negatively.

Furthermore, F_{topic} and F_{senti} are extracted either for each target specifically, or globally: for example, the topic *healthcare* could be a general topic for all politicians or applied to only few politicians based on LDA's topic distribution.

3.3 Temporal Causality Detection

Given a pair of target y and source x time series, we define a causality function $\mathbf{C}$ for calculating causality score between y and x (*Regressor* in Figure 2). 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):

$$y_t = \alpha y_{t-l} + \beta x_{t-l} + \epsilon_t \quad (4)$$

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 number of multi-variate feature

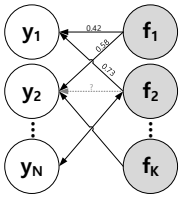


Figure 3: A graphical representation of a bipartite graph G_{TC} between target time series $Y = \{y_1, ...y_N\}$ and feature time series $X = \{f_1, ...f_k\}$ called Temporal-Causality graph (TC-Graph).

time series $X = (f_1, ...f_k)$:

$$y_t = \alpha y_{t-l} + \boldsymbol{\beta} X + \epsilon_t \quad (5)$$

$$= \alpha y_{t-l} + \boldsymbol{\beta}(f_{1,t-l} + ... + f_{k,t-l}) + \epsilon_t \quad (6)$$

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. Then, our causality function $\mathbf{C}$ for y and X is as follows:

$$\mathbf{C}(y, f, l) = \Delta \mathrm{Var}(\beta_{y,f,l}) \quad (7)$$

where Δ is change of variance by the feature f with lag l. Then, the total Granger causality of target y is computed by summing the change of variance over all lags and all features selected by our feature composition function Φ (will be described in the next section):

$$\mathbf{C}(y, X) = \mathbf{C}(y, \{f_1, ...f_k\}) = \sum_{k,l} \mathbf{C}(y, f_k, l) \quad (8)$$

However, the pairwise Granger causality function $\mathbf{C}(y, f, l)$ has a scalability issue when the number of target time series and features are large. For example, we have N target time series from about hundreds companies' stock prices and more than millions K textual features (e.g., F_{words}, F_{topic}, and F_{senti}) collected from text data. A dense matrix of $\mathbf{C}(y, f, l)$ with $N \times K \times L$ size is intractable for

handling in our tasks such as forecasting. Therefore, we represent the dense matrix of $\mathbf{C}(y, f, l)$ into sparse form of bipartite graph G_{TC} called TC-Graph between the target time series Y and the features F (See Figure 3). The TC-Graph is generated by following procedure:

- Insert an edge ($y \xrightarrow{\sum_l \mathbf{C}(y,f,l)} f$) if $\sum_l \mathbf{C}(y, f, l) \geq \xi$ and $\sum_l \mathbf{C}(f, y, l) \leq \xi$
- Insert an edge ($y \xleftarrow{\sum_l \mathbf{C}(y,f,l)} f$) if $\sum_l \mathbf{C}(y, f, l) \leq \xi$ and $\sum_l \mathbf{C}(f, y, l) \geq \xi$
- Insert an edge ($y \xleftrightarrow{\sum_l \mathbf{C}(y,f,l)} f$) if $\sum_l \mathbf{C}(y, f, l) \geq \xi$ and $\sum_l \mathbf{C}(f, y, l) \geq \xi$

where ξ is a confidence threshold for detecting Granger causality. The bipartite representation of our causality model has many advantages in feature composition method described in the following section.

3.4 Feature Composition

Our next goal is to find the best set of features Φ over TC-Graph that maximize our causality function $\mathbf{C}$. We address two practical challenges for feature composition: *noisiness* and *hidden edges*.

The *Noisoness* is very critical problem in practice. Due to the pairwise causality in TC-Graph , the relative importance of Granger causality between features is often disregarded. For example, a very common and general meaning feature $f^{know} \in F_{word}$ or $f^{sports} \in F_{topic}$ has causality with many target time series in fact due to its burstiness. To filter our such burst and noisy features in TC-Graph, we used a modified version of TF-IDF between a feature f and a target y:

$$TFIDF_{y,f} = TF_{y,f} \cdot IDF_f \quad (9)$$

$$= \mathbf{C}(y, f) \cdot \frac{N}{|\{y \in Y : f \rightarrow y\}|} \quad (10)$$

where the term frequency $TF_{y,f}$ is computed by our causality function $\mathbf{C}$ between y and f, and the inverse document frequency is computed by dividing the total number of targets N by the number of targets Granger caused by feature $|\{y \in Y : f \rightarrow y\}|$. Using the TFIDF scheme, we filter out non important causal edges by:

$$G_{TC} = \{e(y, f) \text{ where } TFIDF_{y,f} > \sigma\} \quad (11)$$

where $e(y, f)$ is an edge between target y and feature f, and σ is a threshold for TFIDF filtering.

Another issue is how to find *hidden edges* between targets and features. Since G_{TC} is computed by certain period of time range, a snapshot of the graph may not cover all temporal causality between a target and a feature unless the graph is computed in very long time period. For example in the Figure 3, there is no edge between y_2 and f_2 where it should be. If we know both f_1 and f_2 cause y_1 with strong Granger causality, then we can guess that f_2 possibly causes y_2 because f_1 causes y_2. The basic assumption beyond this is if two features cause the same target, then probably they are semantically similar causes. Based on this assumption, we factorized the bipartite graph G_{TC} into two matrices W and H using non-negative matrix factorization (NMF) (Hoyer, 2004):

$$G_{TC} \approx W \cdot H = G' \quad (12)$$

where the loss function is to minimize $E(W, H) = \sum_{i,j}(G_{i,j} - (WH)_{i,j})^2$. From the approximation of the factorized matrices $W \cdot H = G'$, some of hidden causality values are recovered. For each target y we choose the top features from the factorized G'_{TC} .

$$\Phi(G_{TC}, y) \approx \Phi(G'_{\{f_1,...,f_k\}}, y) = \{f'_{1,y}, ..., f'_{t,y}\} \quad (13)$$

where $\{f'_{1,y}, ..., f'_{t,y}\}$ is a final set of features with highest edge values chosen for target y.

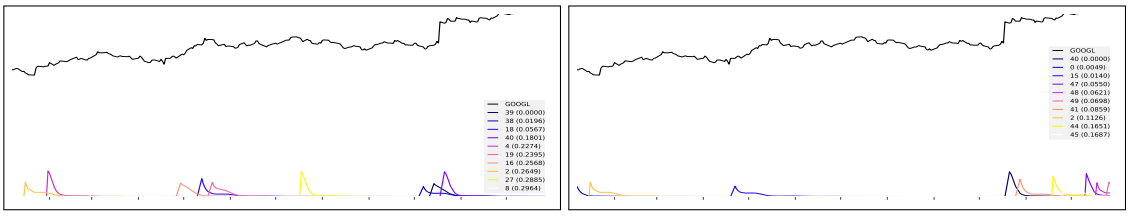
4 Experiments

We first describe data, textual features, and evaluation tasks:

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 Horse 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 [2], Huffington Post [3], Reuters [4], etc. The data covers not only the nomination polling and the presidential polling dated from 2011-02-15 to 2012-11-07, but also President Obama's approval/disapproval polling until present.

Features. For N-gram word features F_{word}, as described in the previous section we choose the spiking words based on their temporal dynamics (See Table 2). For example, if a word is too frequent or the time series is too burst, the word

[2] http://www.usatoday.com
[3] http://www.huffingtonpost.com
[4] http://www.reuters.com



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

Figure 4: 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 with lag size of 3, while (b) shows how the target y causes the random features rf with lag size of 3. The color of random time series r changes from blue to red and yellow according to Granger causality degree with the target (blue is the strongest, and yellow is the weakest). The only top 10 random features with highest Granger score are chosen.

Table 2: Examples of F_{words} with their temporal dynamics: Shannon entropy, mean, standard deviation, maximum slope of peak, and number of peaks.

	entropy	mean	STD	max_slope	#-peaks
spiders	0.46	16.12	9.04	6.33	589
#lukewilliamss	0.72	22.01	18.12	6.12	31
happy_thanksgiving	0.40	61.24	945.95	3423.75	414
christmas_eve	0.46	43.68	480.49	182.53	530
michael_jackson	0.46	141.93	701.97	389.19	585

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 number of words, we finally obtain $21,120$ words as final candidates for F_{words} including unigram 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.

Tasks. To show validation of Granger causality, first we conduct random analysis between target time series and randomly generated time series. For prediction accuracy of our time series model, we test different forecasting algorithms on political poll and stock price tasks. At last, our qualitative result shows the empirical evidence of some interesting causal text features found with Granger causation with target time series.

4.1 Random Causality Analysis

To check whether our causality function $\mathbf{C}(y, rf)$ detects the temporal causality well or not, we conduct a random analysis between target time series t and randomly generated time series rf (See Figure 4). 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)[5].

The color of random time series rf changes from blue to red and yellow according to Granger causality degree with the target $\mathbf{C}(y, rf)$ (blue is the strongest, and yellow is the weakest). The Figure 4 (a) shows what random features rf cause the target y with some lags, while the Figure 4 (b) shows what random features rf are caused by the target y with some lags. We observe that the strong causal (blue colors) random time series 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. That is, if the random time series has high strength of peak pattern, it has higher Granger causality with the target time series even though it is not the closest.

The random analysis shows that our Granger function $\mathbf{C}$ appropriately finds cause and effect relation between two time series in regard of their strength and distance. Also, further experiments shows that depending on the direction of causation (e.g., $\rightarrow$ or $\leftarrow$) or size of lag l, causality changes

[5]SpikeM (Matsubara et al., 2012) has specific parameters for modeling a time series such as peak strength, length, etc. Though, you can use any other spike generation model.

Table 3: Forecasting errors (RMSE) on **Stock** and **Poll** data with different models: time series only models such as *AR*, *SpikeM*, and *NeuralNet*, and time series and text feature models with different types of features such as *random*, *words*, *topics*, *sentiment*, and *composition*. (window size = 30, moving size = 10, lag=3 for **Stock** and lag=1 for **Poll**)

		Time Series			Time Series + Text				
	Step	AR	SpikeM	NNet	$\mathbf{C}_{rand}$	$\mathbf{C}_{words}$	$\mathbf{C}_{topics}$	$\mathbf{C}_{senti}$	$\mathbf{C}_{comp}$
Stock	1	**1.25**	102.13	6.80	3.63	2.97	3.01	3.34	1.96
	3	**2.19**	99.8	7.51	4.47	4.22	4.65	4.87	3.78
	5	**2.91**	97.99	7.79	5.32	5.25	5.44	5.95	5.28
Poll	1	**0.85**	10.13	1.46	1.52	1.27	1.59	2.09	1.11
	3	**1.43**	10.63	1.89	1.84	1.56	1.88	1.94	1.49
	5	**1.68**	11.13	2.04	2.15	1.84	1.88	1.96	1.82

appropriately.

4.2 Forecasting with Textual Features

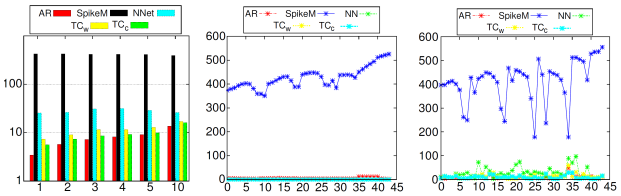
Forecasting is the most interesting application of causality. We use forecasting accuracy as an evaluation metric of whether our textual features are causing the target time series or not. The basic assumption is, if a time series x causes a time series y in the past, x will cause y again in the future so forecasting error rate with good causal time series x would be less. We test our feature composition function Φ by comparing time series only model, and time series and textual feature model.

Two forecasting scenarios are considered: predicting stock price of companies (**Stock**) and predicting poll value for presidential election (**Poll**). For stock data, We collect stock prices during 2013-01-01 to 2013-12-31 only for ten IT companies[6]. For poll data, we choose ten candidate politicians [7] 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.

For forecasting models only with past time series data, we used three regression based algorithms: Auto-regression (AR) (Hamilton, 1994), SpikeM (Matsubara et al., 2012), and NNet (nne, 2015). The AR fits the past time series using simple regression model $x_t = \sum x_{<t} + \epsilon$. The SpikeM models a time series with a handful number of parameters such as strength, volume, noise, etc. The NNet is simple LSTM (Hochreiter and Schmidhuber, 1997) based time series model.

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

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



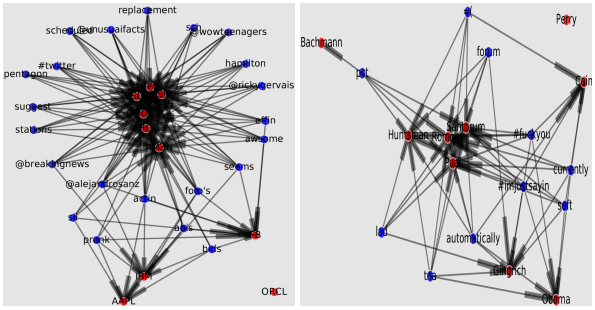
(a) Forecasting error per step size **(b)** Fitting error by moving windows **(c)** Forecasting error by moving windows

Figure 5: Forecasting analysis: (a) forecasting error per step size, (b) fitting error by moving windows, and (c) forecasting error by moving windows (windows=30, moving size=5).

For forecasting models with past time series as well as textual features, we use Vector AutoRegressive model with eXogenous variables (VARX) (Hamilton, 1994) with different types of 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.

Table 3 shows forecasting errors 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 by moving a window (size = 30 days) 10 days over the period. Among the time series only models, *AR* outperforms *SpikeM* and *NeuralNet*. Especially, *SpikeM* is regressive on long term peak patterns that is not appropriate with our window based evaluation. Among the time series and textual feature models, $\mathbf{C}_{composition}$ outperforms all other features, indicating that our feature selection method for filtering out meaningless features and finding hidden edges is effective for choosing the best set of features. However, it is still difficult to beat the time series only models because our window based experiment is limited to miss detecting the long range of effect derived from external features. The long range analysis with the compositional causative features will be studied in our following work.

Figure 5 (a) shows forecasting errors with different step sizes. As the step size increases, the gap between AR model (the best in time series only models) and our composition model decreases, indicating that our textual features help forecast longer range of patterns. The fitting (b) and forecasting (c) errors through the time win-



(a) Stock (*Degree* > 5) **(b)** Poll (*Degree* > 5)

Figure 6: TC-GRAPH with features whose degree is higher than five. The red circles are either companies or politicians, while the blue circles are features.

dows show that how difficult forecasting is compared to fitting.

Table 4: Causal features for IT companies' stock price change during 2013.

Amazon	Google	Facebook	Apple
xbox	more_productive	irespectfemales	election_day
brett_favre	#idoit2	beautiful_people	of_november
operating	say_thank	six_pack	the_packers

4.3 Qualitative Analysis

Figure 6 shows visualization of our TC-GRAPH by filtering out some nodes based on the graph degrees: (a) and (c) show feature whose degree is higher than five, and (b) and (d) show feature with degree equal to one. The visualization shows how companies and politicians are mutually caused by certain features, respectively. Interestingly, "awesome" feature is shared in many tech companies like Amazon and Google but not in Oracle. In (b), IBM is somehow causally related to "@arsenal". Table 4 shows some causal features for IT companies. For example, the stock price of *Amazon* has high causal relation with the word features "xbox", "brett_farve" or "operating".

5 Conclusion

We extracted textual features from online media and found temporal causality with target time series such as companies' stock prices and politicians' election polls. Our proposed method generates a bipartite graph whose edges are Granger causalities, filters some noise features and finds some meaningful but hidden nodes. We tested random causality analysis, forecasting, and qualitative analysis on the detected causality.

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 .

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* .

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*, Springer, pages 177–190.

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

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*. Association for Computational Linguistics, pages 294–303.

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*. Association for Computational Linguistics, pages 76–83.

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

Clive WJ Granger. 1992. Forecasting stock market prices: Lessons for forecasters. *International Journal of Forecasting* 8(1):3–13.

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

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.

Maciej Kamiński, Mingzhou Ding, Wilson A Truccolo, and Steven L Bressler. 2001. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological cybernetics* 85(2):145–157.

Dongyeop Kang, Varun Gangal, and Eduard Hovy. 2017. Generating causal explanations with symbolic and neural reasoning. *preprint, "http://www.cs.cmu.edu/~dongyeok/papers/preprint/kang17explanation_acl_submission.pdf"* .

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

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.

Paramita Mirza. 2014. Extracting temporal and causal relations between events. *ACL 2014* page 10.

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

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM* 11(122-129):1–2.

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*. IEEE, pages 1074–1079.

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* .

Reinhard Schwarz and Friedemann Mattern. 1994. Detecting causal relationships in distributed computations: In search of the holy grail. *Distributed computing* 7(3):149–174.

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*. Association for Computational Linguistics, pages 347–354.