Detecting Evolution of Concepts based on Cause-Effect Relationships in Online Reviews

Yating Zhang, Adam Jatowt and Katsumi Tanaka
Graduate School of Informatics
Kyoto University, Yoshida-honmachi, Sakyo-ku
606-8501 Kyoto, Japan
{zhang, adam, tanaka}@dl.kuis.kyoto-u.ac.jp

ABSTRACT
Analyzing how technology evolves is important for understanding technological progress and its impact on society. Although the concept of evolution has been explored in many domains (e.g., evolution of topics, events or terminology, evolution of species), little research has been done on automatically analyzing the evolution of products and technology in general. In this paper, we propose a novel approach for investigating the technology evolution based on collections of product reviews. We are particularly interested in understanding social impact of technology and in discovering how changes of product features influence changes in our social lives. We address this challenge by first distinguishing two kinds of product-related terms: physical product features and terms describing situations when products are used. We then detect changes in both types of terms over time by tracking fluctuations in their popularity and usage. Finally, we discover cases when changes of physical product features trigger the changes in product's use. We experimentally demonstrate the effectiveness of our approach on the Amazon Product Review Dataset that spans over 18 years.

Keywords
Technology evolution analysis, product evolution analysis, social influence, causality detection.

1. INTRODUCTION
Knowledge of product and technology evolution can offer many advantages. First, it can help people to learn about the way in which innovations appear and evolve. Users can gain insight into the way in which the changes of product features impact changes in their daily lives. Second, producers when equipped with comprehensive information on how products evolve could make better decisions. Finally, scientists in the areas related to the social aspects of technology and the history of science could benefit from data-driven approaches to support different hypotheses or verify their discoveries.

Although "evolution" has been already explored in many subdomains of computer science, such as the evolution of topics [1,9,36], named entities [21,34] and terminology [3,15,16,20], few works approached the problem of automatically analyzing and portraying the evolution of products and technology. Yet, at the same time, a large number of product reviews have been created by users in the progress of the last years. Hence, collections of product-related documents that span multiple years are already available making it possible to conduct various kinds of temporal analyses including the studies of technology evolution.

In this work, we propose to use the collections of product reviews for automatically analyzing product evolution. However, the type of the evolution analysis that we conduct is novel and diverts from the typical studies of topic or event evolution [1,6,9,36]. We believe that at least two points are important in the technology evolution analysis, and, which, in our opinion, have not been sufficiently studied within the realm of data mining and computer science.

First, the evolution of products is a socially-related construct. The features of commonly used products usually have strong impact on the society. For example, it is known that early portable music devices equipped with batteries and earphones were small and efficient enough to let their users freely listen to music while performing outdoor sports. Based on the above example we could say that product features (e.g., `earphones` and `batteries`) had certain social impact on user lifestyle (e.g., `performing outdoor sports`). In general, we think it is important to detect not only novel product features appearing over time but, more interestingly, to pinpoint their actual social impact in order to exhaustively reflect the evolution of products and technology in general. Actually, the phenomenon of technology impact on society has been an interest of social scientists and historians for long time [4,7,27]. The concept of *technological determinism* [7] presumes that a society's technology drives the development of social structure and cultural values. We thus put special emphasis in this work on social aspects of technology progress.

Second, although the straightforward statistical analysis [29,33] could explain the changes of a single product, generalizing to entire product categories is difficult. In other words, generic models that could be flexibly applied over different scopes (or product categories) to analyze the technology evolution are difficult to construct. To give an example, there should be a way to not only portray the evolution of `Walkman` (a particular product), but also the evolution of portable music devices (a product category) or music devices (a larger category) in general. Such flexible approach would allow for comparison of the evolution occurring across different granularity levels and hierarchies.

We address the above mentioned challenges by providing novel methodology to extract evidences of technology impact on human activities. As the underlying document collection we use the Amazon Product Review Dataset [22] which contains over 34 million reviews about over 2.4 million products written by 6.6 million users from June 1995 to March 2013.

We formulate our task as the detection of cause-effect relations described in text, such that the change of product features causes (or influences) the changes in lifestyle. We first propose two types of time series (frequency-based and semantic-based) to capture the

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
DOI: http://dx.doi.org/10.1145/2872427.2883013.

change of terms denoting product features, social actions and usage in technology. We detect changes of these terms over time by tracking fluctuations in their popularity and shifts in their context. For capturing the latter we employ neural networks which are sequentially retrained on temporal subsets of our dataset. Next, we propose methods for detecting causal relations based on both the time series and term probabilities in text. Furthermore, we go beyond detecting only binary causal relations attempting at finding the concept-level causality by aggregating binary causal relations. Note that, for longitudinal temporal document collections such as the one we use, the standard Natural Language Processing (NLP) methods [11,26,30] designed for causality or entailment detection within text cannot be directly applied. These can extract only explicitly mentioned causal relations, while in our work, we aim to detect implicit causality that relate temporally distant events.

To sum up, we make the following contributions in this paper:

1. We describe a novel approach for analyzing technology evolution by considering its social influence over time.
2. We propose using product reviews collected over multiple years for causality detection. The proposed methods are flexible over different levels of product categories.
3. The effectiveness of our approach is demonstrated by experiments on the Amazon Product Review Dataset spanning 18 years.

The remainder of the paper is organized as follows. We begin in Section 2 with the review of related work. In Section 3, we give the formal problem definition. Section 4 outlines our proposal for constructing time series to model technology changes and changes of product usage. Next, in Section 5, we describe how to uncover causal relations that underlie these changes. The experiments are explained in Section 6 and the evaluation results are discussed in Section 7. We conclude the paper in Section 8.

2. RELATED WORK

2.1 Social Studies of Technology

Relationship between technology and social life has been an important topic of study for sociologists and historians of engineering, technology and science. Advocates of social constructivists, Bijker *et al.* in their book "The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology" [4] emphasized the concept of *Social Construction of Technology* (SCOT). Their theories support the key idea of SCOT that technology does not determine human action, but that, rather, human actions and human needs shape the evolutionary paths of technology. In contrast to SCOT, *technological determinism* [7] presumes that society's technology drives the development of social structures and cultural values. The notion of the interaction between the technology and society underlies the key motivation behind our work that understanding product evolution is a way to better understand the changes of society.

2.2 Product Evolution

When it comes to data mining approaches for the automatic analysis of product evolution the prior literature is rather sparse. Radhakrishnan *et al.* [29] researched modeling product evolution focusing on automatically ordering the names of product models (e.g. Windows 95>98>2000>XP>7.0>8.0) by training CRF model using several classifier features. Strötgen *et al.* [33] explored relations among products by utilizing simple statistical analysis such as the changes in number of reviews and ratings.

Our work differs from these researches in several aspects. First, we introduce a novel viewpoint to look at the evolution of products through the "social impact lens" for understanding how changes in product features impact the changes in human life. Second, we propose an approach to explore and quantify causal relationships. Third, our work is not limited to any specific models or to any category level, so the proposed approach can be flexibly applied over different levels of categories (e.g., switching from the category of music devices to the lower category of mp3 players).

2.3 Causality Detection

A variety of approaches from computer science and statistics have been developed for detecting causal relations [18]. These can be categorized into NLP approaches [11,26,30], graphical models [8,32], Granger causality [2,10,12] and temporal logic [13,19].

Most of the causality works in the area of NLP are based on pattern extraction [11,26,30]. For example, Girju *et al*. [11] proposed detecting causality by first discovering the lexico-syntactic patterns in text referring to causation and then applying several semantic constraints to validate and rank the candidate patterns using the confidence scores generated by these constraints. Other research works such as [26,30] focused on the problem of predicting the causality between events by generating semantic rules to express the causality in the text. However, the limitation of the semantic rule (or pattern) based approach is difficulty of extracting all possible ways in which users could express causality. Another disadvantage of that approach is that it can extract only explicitly mentioned causal relations in text, while in our work, we attempt to detect implicit causality that relates temporally distant events, and it is not directly referred to in text.

Graphical models, such as Bayesian Networks (BN) and Dynamic Baysian Networks, infer probabilistic models representing the set of causal relations between variables. The main idea is to find a graph or a series of graphs that best explain the data. The methods of BN inference can be further divided based on the way to find the best graph: one is to search over the set of possible graphs to maximize a scoring function [8], the other is to start with a fully connected graph and remove the edges by conducting conditional independence tests [32]. This approach is however limited due to its significant computation complexity which makes it difficult to search exhaustively over all possible graphs.

Granger causality [12] is a popular approach for causal relation detection, and is, especially, used in economics. The idea is to perform hypothesis test for every pair of features on whether one time series of a feature is predicative about the time series of another feature within a given lag. In other words, a variable A causes B if A provides extra explanatory power to predict B more than the past values of B itself. However, this approach has several drawbacks. It involves applying regression on lags resulting in $O(p^2)$ time, where p is the number of features. Also, the statistical significance tests are conducted sequentially for all pairs of features. Several extension works based on the concept of Granger causality have been proposed to tackle the complexity problem, such as Lasso Granger method [2], Vector Auto-Regressive (VAR), [10] etc. Another drawback is that Granger causality based methods as well as Bayesian Nets typically solve the problem of *type-level causality*, which targets periodical or general causal relations (e.g., low air pressure causing rain, or common cold causing runny nose). However, our goal is to detect *token-level causal relations* where the cause is specific and often occurs only once (e.g., the release of a particular novel technology causing certain user response). In general, *token-level causality* is more difficult to be detected than *type-level causality* due to the lack of direct prior evidences.

The approach based on temporal logic [13,19] can be flexibly used to test unstructured relationships for specific time periods.

This approach is based on the assumption that a cause precedes its effect and the cause raises the probability of the effect [13]. Since temporal logic is based on probability, it can be directly employed for analyzing *token-level causality*.

We thus adopt the temporal logic approach in this work thanks to its capability of handling the *token-level causality*. However, unlike the previous solutions [13,19], our work is different in that (1) it detects implicit causal relations between words within temporal document collections. In particular, it extracts word level causal relations by causally binding changes related to one word with the changes related to another word within a specific time period. (2) Discovering causality between words is difficult due to the lack of direct prior evidences. So, in addition to detecting causal relations between two words, we also provide aggregation ways to better estimate the causal strength by grouping similar concepts or similar causal patterns.

3. PROBLEM STATEMENT

We first introduce the formal description of the problem.

Causal relation, $c \rightarrow e$, is the relation between a cause c and an effect e, such that the change in the cause leads to the change in the effect. In this work, we specifically allow the cause to be one of physical product features and the effect to be one of usage words describing the way in which products are used. To detect physical product features we employ classification approach (described in detail in Sec. 6.2), while we utilize verbs (e.g., `download`, `jogging`) and situation terms such as locations (e.g., `gym`, `home`) as terms denoting the product use.

Formally, the strength of *causal relation*, $I(c \rightarrow e)$, is defined as:

$$I(c \rightarrow e) = f(d(c), d(e))$$

where function d (described in Sec. 5.1) quantifies the change of a cause term c and the change of an effect term e. Function f (described in Sec. 5.2-5.3) estimates the causal strength between the two change types.

The output consists of the ranked lists of detected causal relations as well as the time of their occurrences. Note that to allow for flexible approach the input data can be any category level of products.

4. TIME SERIES CONSTRUCTION

Constructing time series is the first step for detecting significant changes related of words. In this section, we introduce two approaches to measure the change, *frequency-based* and *semantic-based*. The former tracks the changes of word popularity over time, while the latter quantifies variations in word usage.

4.1 Frequency-based Approach

Tracking the frequency of a word over time is a simple method to find time periods when the word has been increasingly used. In case of product reviews these often indicate time when new innovations came or new means of product usage occurred. To measure term frequency we split the dataset into non-overlapping time units and we calculate the average frequency of a term occurrence per document at each time unit as follows:

$$Freq_t(w) = \frac{\#(w \in D_t)}{|D_t|} \quad (1)$$

where $\#(w \in D_t)$ is the number of occurrences of a word w in the document set D_t at time unit t. $|D_t|$ is the number of documents published at t.

We note that the frequency-based method can be easily implemented and can scale up to big datasets. It is also relatively easy to measure correlations of two words by detecting their co-occurrences in documents.

4.2 Semantic-based Approach

While the frequency measure is easy to implement, it cannot capture cases when a word is used with stable frequency, yet, its surrounding context changes. For example, in the category *Electronics, Portable Audio & Video*, the frequency of the term `car`, which is considered here a situation word (description of a place where products can be used), does not change much across time in the Amazon Product Review Dataset. Yet, its context changes a lot. One reason for this is the change in the types of music devices used while travelling by car, which evolved from cassette-based, through CD-based to mp3-based ones. This kind of semantic shifts can be detected by analyzing fluctuations in distributed representation of words. Distributed representation of words is based on the *distributional hypothesis* which states that words appearing in similar contexts are semantically similar. Distributed representation enables to represent the semantics of a word by analyzing its context and to measure the semantic similarity of words as the distance between their vectors. Computing such representation by neural networks was first proposed by Rumelhart *et al.* [31] in 1986. More recently, Mikolov *et al.* [24,25] introduced the Skip-gram model which utilizes a simplified neural network architecture for learning vector representations of words from unstructured text data. We apply this model due to the following advantages: (1) it can capture precise semantic word relationships; (2) the model can easily scale to millions of words due to the simplified neural network architecture.

Fig. 1 overviews the process of constructing semantic time series of words. First, we collect all the words that occurred higher number of times than the predetermined threshold (5 times) at any time within our dataset. Based on these vocabulary we then train the Skip-gram[1] model using all the reviews published in the first year. Thus from the beginning, every word is going to have a position in the vector space. Note that, for those terms which have not appeared in the first years, the model still assigns some initial vectors. Next, for each subsequent time unit[2], we iterate over epochs and train the word vectors until convergence. We define the measure of convergence as the average angular change in word vectors between epochs as shown below:

$$\rho = \frac{1}{|V_t|} \sum_{w \in V_t} \arccos \frac{\chi_w(t,\omega) \cdot \chi_w(t,\omega-1)}{\|\chi_w(t,\omega)\|_2 \|\chi_w(t,\omega-1)\|_2} \quad (2)$$

where the $\chi_w(t,\omega)$ is the vector of word w at time slot t and epoch ω. For each time slot t, after each epoch, the model will stop updating the word vectors if ρ is lower than 0.0001.

We finally construct the time series of a word w by computing the semantic distance between its distributed representations at time t and at time *t-1* (see Fig.1). The semantic distance is measured as follows[3]:

$$Dist_t(w) = 1 - \frac{\chi_w(t) \cdot \chi_w(t-1)}{\|\chi_w(t)\|_2 \|\chi_w(t-1)\|_2} \quad (3)$$

[1] We use a window size of 10 and the dimension number of 100.

[2] We use one month as a time unit in the experiments.

[3] Note that there is no need for applying space transformation such as the one in our previous work [35] because term vectors at each time unit are computed by retraining data from the previous time unit.

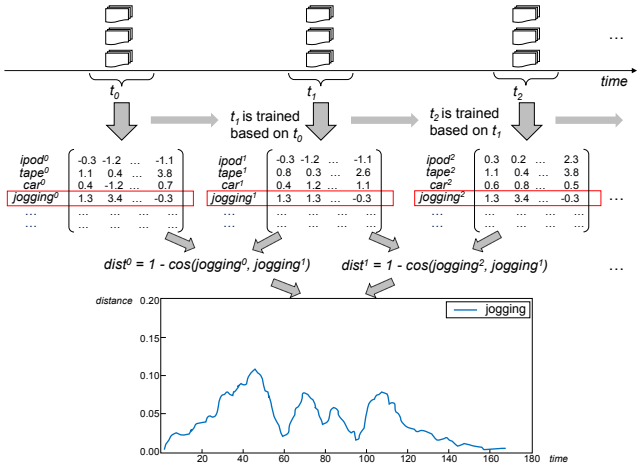


Figure 1. Overview of Constructing Semantic-based Time Series.

5. CAUSALITY DETECTION

As discussed in the problem statement (see Sec. 3), we consider the occurrence of the changes related to physical product features and the ones related to terms describing product usage as the premise of the occurrence of any causality. In this section, we first construct function *d* for detecting changes in time series. We then create function *f* by estimating the causality strength between pairs of a change in physical product features and the one in product usage. Finally, we propose two ways for aggregating binary causal relations.

5.1 Change Detection

Changes in the two time series representations of words (frequency-based and semantic-based) discussed in Sec. 4 have different meanings. When the frequency of a word significantly increases, it means the word becomes more popular which may be the result of either some change in technology or change of concepts related to or dependent on the technology. On the other hand, in the case of the semantic-based time series, each point actually represents the dissimilarity (distance) when compared with the data from the previous time unit. Therefore, in this case, the peak points (or points above threshold) mean high context change. Similarly, valley points (or points below threshold) indicate little context change or the lack of any change. Thus, we can regard the peak periods as semantic change periods.

In the frequency-based time series we detect the increase periods by assuming that the increase period is the period between the adjacent "valley" and "peak". To detect peaks and valleys, we refer to the method described in [5], the idea of which is that a peak is considered as the highest point between "valleys". What makes a peak is the fact that there are lower points around it. We employ this method due to its advantage of fast computation and flexibility in parameter changes. It uses the distance, denoted as *lookahead*), to look ahead from a peak candidate. A minimum difference (denoted as *delta*) between a peak and the following points is also specified to distinguish an actual peak from a jitter. After detection of peaks, we estimate the increase period by considering a parameter *slope*, which guarantees the absolute value of minimum slope within the increase period.

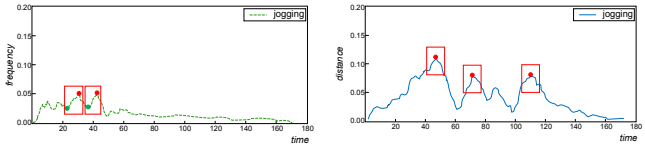
Algorithm 1 describes the process of the change detection[4] over both the frequency-based and the semantic-based time series in detail. It is composed of two parts. For finding the frequency changes we compute part 1 and 2 (the increase detection is based on peak detection), and for finding the semantic changes we only compute part 1 (peak detection).

Algorithm 1 ChangeDetection($\mathcal{T}$)

Input: Time series $\mathcal{T}$, Parameters *lookahead*, *delta*, *slope*
Output: Change periods ($Peak_{\mathcal{T}}$ or $Increase_{\mathcal{T}}$)

```
1:  /* Part 1: Peak and Valley Detection */
2:  minima, maxima ← ∞, -∞
3:  for T_i in T do:
4:    if T_i > maxima then
5:      maxima ← T_i
6:    end if
7:    if T_i < minima then
8:      minima ← T_i
9:    end if
10:   if T_i < maxima − delta and maxima ≠ ∞ then
11:     if maxima > {T_k | T_k ∈ T_[i, i+lookahead]} then
12:       Peak_T ← add (T_i, maxima)
13:       maxima, minima ← ∞, ∞
14:     end if
15:   end if
16:   if T_i > minima + delta and minima ≠ -∞ then
17:     if minima < {T_k | T_k ∈ T_[i, i+lookahead]} then
18:       Valley_T ← add (T_i, minima)
19:       maxima, minima ← -∞, -∞
20:     end if
21:   end if
22: /* Part 2: Increase Detection */
23: for adjacent (valley, peak) in (Valley, Peak) do
24:   leftbound ← (T_peak − T_leftbound) / (peak − leftbound) > slope
25:   Increases_T ← add (leftbound, peak)
26: end for
```

Fig. 2 shows the example of change periods detected for the term `jogging` in its frequency-based time series (Fig. 2(a)) and the semantic-based time series (Fig. 2(b)).



(a) Frequency-based time series (b) Semantic-based time series

Figure 2. Example of change periods detected for the term `jogging` for the frequency-based (Fig. 2(a)) and the semantic-based (Fig. 2(b)) time series. Green dots indicate the valleys of the time series, and the red dots represent the peaks of the time series. Red rectangles mark the detected change periods.

5.2 Estimating Causal Strength

To measure the strength of causality between a cause and an effect we adapt the approach of temporal logic [17,18]. It is based

[4] We use value of 18 for *lookahead*, 0.01 for *delta* for both for frequency- and semantic-based time series. *slope* is equal to 0.005 for the frequency-based time series for increase detection.

on the key principles of probabilistic causality that: (1) a cause temporally precedes its effect, (2) the occurrence of the cause raises the probability of its effect.

Based on these principles, Eq. 4 gives an estimation of causal strength between a cause c and an effect e. Intuitively, it can be interpreted as the more the probability of the occurrence of an effect e increases given the occurrence of cause c, the higher is the strength with which c causes e. The causal strength is associated with a causal time period [ts, te] which is necessary considering that the same causal relation (i.e., relation binding the same pair of a cause and an effect) can have different causal strengths at different time periods:

$$I_{ini}^{[ts,te]}(c,e) = P_{[ts,te]}(e \mid c) - P_{[ts,te]}(e \mid \neg c) = \frac{tf(e \in M_{[ts,te]}(c))}{\sum_{i \in V} tf(i \in M_{[ts,te]}(c))}) - \frac{tf(e \in M_{[ts,te]}(\neg c))}{\sum_{i \in V} tf(i \in M_{[ts,te]}(\neg c))} \quad (4)$$

where $P_{[ts,te]}(e|c)$ is the probability of an effect e given the occurrence of cause c within [ts, te]. We compute this probability by dividing the term frequency of e in the documents which contain the cause c by the one in the documents that lack c. $M_{[ts,te]}(c)$ is the set of documents which contain c during the time period [ts, te] and $M_{[ts,te]}(\neg c)$ is the set of documents which do not contain c during [ts, te]. V represents all terms which appear in $M_{[ts,te]}(c)$ or the ones in $M_{[ts,te]}(\neg c)$.

One problem with Eq. 4 is that it is computed irrespectively of the influence of other potential causes on the same effect. In particular, there may exist several other causes of the effect (for example, ones causing both c and e). In another case, c may not be a genuine cause (being only a spurious or a weak cause). Thus, based on the list of candidate causes computed by Eq. 4, we determine whether a particular candidate cause c is a significant cause of e by contrasting it with all the candidate causes of e. The global causal strength of $c \rightarrow e$ is then computed by Eqs. 5 and 6.

$$I_{glob}^{[ts,te]}(c,e) = \frac{\sum_{x \in X \setminus c} \varepsilon_x^{[ts,te]}(c,e)}{|X \setminus c|} \quad (5)$$

where,

$$\varepsilon_x^{[ts,te]}(c,e) = P_{[ts,te]}(e \mid c \wedge x) - P_{[ts,te]}(e \mid \neg c \wedge x) = \frac{tf(e \in M_{[ts,te]}(c \wedge x))}{\sum_{i \in V} tf(i \in M_{[ts,te]}(c \wedge x))}) - \frac{tf(e \in M_{[ts,te]}(\neg c \wedge x))}{\sum_{i \in V} tf(i \in M_{[ts,te]}(\neg c \wedge x))} \quad (6)$$

Here, X is the list of candidate causes obtained by applying Eq. 4. Intuitively, Eqs. 5-6 estimate the causal power of a given candidate cause c by measuring how much, on average, the co-occurrence of c with other candidate causes can explain the appearance of e. Note that using only the selected candidates helps to reduce the computation time of Eqs. 5 and 6.

5.3 Aggregation of Causal Relations

The above-described method outputs binary causalities, that is, it computes causal strength between pairs of a single cause and a single effect. However, binary causal relation has limited power to prove the credibility of the findings. We thus propose two aggregation ways to accumulate the influences of other binary causations which have similar meaning to the target pattern. In other words, by aggregation, we can find more evidence to support particular result. For example, if the following binary causal relations are returned `mp3→walking`, `minidisc→running` and `mp3→jogging`, then we are more confident in their correctness since they essentially mean similar things, and thus support each other. Intuitively, we would like to merge such pairs and combine their importance measures.

The objective of the extended method is then to discover and aggregate causal relations that are semantically similar. Since there are two sides, cause and effect, we could (*a*) form semantic groups in each side, respectively, or (*b*) construct semantic grouping simultaneously on both sides. These two approaches are described in the following two subsections.

5.3.1 *Grouping by Similar Concepts*

Fig. 3 illustrates the aggregation process by grouping similar concepts. The left graph is the original graph in which thin, grey undirected links mean the connected nodes are semantically similar to each other, while the red directed links denote cause-effect relations. The right hand side graph shows two steps for completing the aggregation: step 1 is to detect clusters of similar causes and similar effects; step 2 is to aggregate the scores of causal strengths between any two clusters. These two steps are discussed in the following.

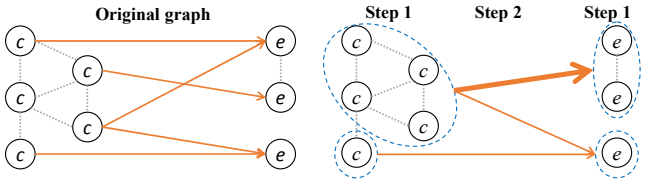


Figure 3. Overview of Aggregation by Similar Concepts.

Step 1. We start with the clustering method to discover semantic groups on the cause and effect sides separately. Take the cause side as an example. We first construct a similarity graph for the causes. Any two causes will be connected in such a graph if their semantic similarity is above the threshold (the threshold equals 0.5 by default). Then we sort all the pairs of connected causes by their semantic similarity. In the process of grouping, we start from the most similar pair (c_i, c_j) to find the common neighbors of c_i and c_j and we group c_i and c_j with such common neighbors. Next, the second least similar pair is taken and the processing goes in the same way. The grouping continues until reaching the least similar pair. In this process, each node may be assigned to different groups. Since the grouping starts from the most similar pair, it thus forms groups with high inner-similarity within the group members, which guarantees the generated concept (or cluster) to be correct and pure. Another advantage compared to other clustering methods is that we do not need to pre-determine the number of clusters as in k-Means (k clusters) or Hierarchical Clustering (degree of *cut*). We conduct the same grouping process on the effect side as well. Algorithm 2 describes the grouping procedure in detail.

Step 2. After grouping cause words and effect words respectively, the final score of causal strength between a cause cluster and an effect cluster is the sum of the global implication scores between the clusters' members normalized by the total number of possible links between both the clusters. The aggregated implication between a group of causes, C and a group of effects, E, is computed in Eq. 7:

$$I_{concept}(C,E) = \sum_{p \in P, c \in C} I_{glob}^{[ts,te]}(c,e) \times \frac{\sum_{c \in C, e \in E} \Gamma(I_{glob}^{[ts,te]}(c,e))}{num(C) * num(E)} \quad (7)$$

$$\Gamma(I_{glob}^{[ts,te]}(c,e)) = \begin{cases} 1 & if\ I_{glob}^{[ts,te]}(c,e) > \sigma \\ 0 & otherwise \end{cases}$$

where σ is the threshold for deciding whether we create a cause-effect relation link between a cause word and an effect word (σ equals 0 by default). $num(C)$ and $num(E)$ are the number of terms in C and the one in E, respectively.

Algorithm 2 GroupSearch(G)

Input: Similarity graph of causes G_{cause} (or effects G_{effect})
Output: Cause groups $C_{group} = \{C_0, C_1, \ldots, C_m\}$
(or Effect groups $E_{group} = \{E_0, E_1, \ldots, E_n\}$)

1: *Pairs* $\{(c_i, c_j), (c_i, c_k), \ldots, (c_j, c_k)\} \leftarrow SortSimilarity(G_{cause})$
/* sort the pair of terms by their semantic similarity from highest to lowest */
2: C_{group} $\{C_0 \supset (c_i, c_j), \ldots, C_u \supset (c_j, c_k)\} \leftarrow IniGIdx(Pairs)$
/* initialize group index */
3: $CurrGIdx \leftarrow 0$
4: **for all** *pair* (c_i, c_j) in *Pairs* **do**
5: $\{c_k, c_p, \ldots, c_q\} \leftarrow CommonNeighbors(c_i, c_j)$
6: **for** c in $\{c_k, c_p, \ldots, c_q\}$ **do**
7: $GIdx(c_i, c_j, c) \leftarrow Minimum(CurrGIdx, GIdx(c_i), GIdx(c_j))$
8: **end for**
9: $CurrGIdx \leftarrow CurrGIdx + 1$
10: **end for**

5.3.2 Grouping by Similar Patterns

In this section, we propose to aggregate causalities by simultaneously considering the similarity on both sides. The idea is that a cause-effect relation *(c→e)* for which there are many similar cause-effect relations, should receive more votes. In other words, such causal pattern is important. Fig. 4 illustrates the aggregation process. The left hand side graph is the initial graph in which a node is a cause-effect relation and an edge represents the semantic similarity between any two cause-effect relations. The right hand side graph is obtained after aggregating the causality scores based on the similarity relations and then grouping causes and effects.

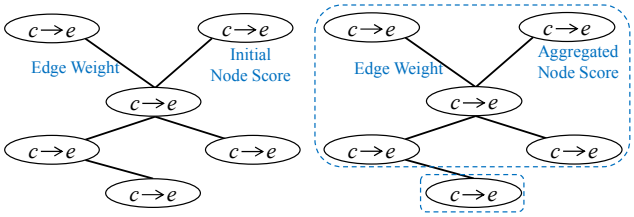


Figure 4. Overview of Aggregation by Similar Patterns.

Formally, let $G = (V, Q)$ be an undirected graph with the set of vertices V and set of edges Q. We put each initial causal relation $(c \rightarrow e)$ that was found so far into a single node and we assign the initial score of the node to be equal to the causal strength of this relationship (as computed by Eq. 5). An edge in G represents the fact that any two causal relations (V_i, V_j) are similar more than the predefined threshold (by default equal to 0.5). An edge weight is estimated by the sum of the semantic similarities of both sides (cause and effect) and is denoted as w_{ji}. Let $Neigh(V_i)$ be the set of vertices that link to a vertex V_i. The score of a vertex V_i, denoted as $Aggr(V_i)$, is computed in a way similar to TextRank algorithm [23] by interactively computing Eq. 8 until convergence (i.e., the difference of aggregated scores is less than 0.0001):

$$Aggr(V_i) = (1-d) + d * \sum_{V_j \in Neigh(V_i)} \frac{w_{ji}}{\sum_{V_k \in Neigh(V_j)} w_{jk}} Aggr(V_j) \quad (8)$$

where d is a damping factor set by default to 0.85.

After aggregating the causality score of each node, we perform a simple node grouping by combing the neighbors of each node and keeping the largest clusters. For example, in case of two clusters $\{V_1, V_2, V_4\}$ and $\{V_1, V_4\}$, we keep the larger cluster $\{V_1, V_2, V_4\}$. Since each node is a causal relation $(c \rightarrow e)$, we combine all the causes and combine all the effects within each cluster to form a concept representing the group of causes, C, and the one representing the group of effects, E. The causality strength between C and E is estimated by the maximum score among all the pairs of cause and effect within C and E (see Eq. 9)

$$I_{pattern}(C, E) = \max_{c \in C, e \in E} \{Aggr(c \rightarrow e)\} \quad (9)$$

6. EXPERIMENTS

6.1 Dataset

For the experiments we use the Amazon Product Review Dataset provided by Stanford Network Analysis Platform (SNAP) [22]. This dataset includes product descriptions as well as their hierarchical categorization and 34 million reviews written for over 2.4 million products. The reviews were written since June 1995 up to March 2013. This period is long enough for evolution studies considering that technology experienced dramatic changes within the recent years. For the evaluation, we choose two large subcategories of *Electronics* category: *Electronics, Portable Audio & Video* and *Electronics, Camera & Photo*. We have selected these categories because many important changes occurred within the past decade in relation to the products belonging to these categories. Table 1 summarizes the statistics of the selected categories.

Table 1. Statistics of Evaluated Categories.

Category	#Products	#Reviews	Time Span
Electronics, Portable Audio & Video	7,809	182,831	2000-2012
Electronics, Camera & Photo	21,289	289,956	1999-2012

6.2 Feature Extraction

First, we need to extract candidate cause and effect terms as briefly mentioned in Sec. 3. In the current implementation, we use verbs (e.g., `navigate`, `scroll`) and situation terms (e.g., `gym`, `home`) as effect terms. We automatically detect verbs using POS tagger[5]. As for the situation terms, we extract the nouns appearing directly after prepositions (e.g., `in`, `during`). Note that it is also possible to utilize existing vocabulary lists for selecting situation terms such as the lists of locations.

On the other hand, automatically retrieving physical product features requires more processing. The problems lie mainly in that: (1) physical product features vary across different products, therefore, we cannot use any predefined or general feature lists while still not missing any unique characteristics of products. Another issue is that (2) some physical terms are novel (e.g., `USB`, `mp3`) which may not exist in commonly used vocabulary lists or dictionaries. It is thus difficult to define the meaning of such new words by using existing dictionaries.

Considering these challenges, we propose classification model to distinguish if a term is a physical product feature or not. As classifier features, we make use of both the semantic meaning of words and their characteristics derived from a lexical database

[5] The following part of speech types were used: VB, VBD, VBG, VBN, VBP and VBZ by applying NLTK POS tagger (http://www.nltk.org/api/nltk.tag.html accessed on 29/01/2016)

such as WordNet[6]. The types of classification features we use are as follows:

1. *Word Semantics*: We use the distributed representation [24] of a word to represent its meaning because similar terms should belong to the same class. To capture term meaning, we train fixed word embeddings by using all the reviews of the target category regardless of their time stamps. After training, similar terms, such as `cassette`, `CD` and `mp3`, will be positioned close to each other in the vector space (since they all represent the same concept of storage medium). This means they are more likely to belong to the same class. We set the number of dimensions for word embeddings to 100.
2. *Lexical characteristics: We select 5 classifier features derived from the WordNet as follows.*
   a. *Distance to the node* "*physical entity*" *(integer)*: We measure the distance to the node "physical entity" in the WordNet hierarchy. The smaller the depth is, the more physical the word is likely to be.
   b. *Distance from the node* "*abstraction*" *(integer)*: We measure the distance from the node "abstraction" to the target word within the WordNet hierarchy. The higher the distance, the more "physical" (less abstract) the word is likely to be. This is because physical product features tend to be represented by concrete, rather than, abstract words.
   c. *Number of hyponyms (integer)*: General words tend to have on average more hyponyms. Since physical product features are usually more specific, then we assume that the fewer hyponyms a word has, the more physical it is likely to be.
   d. *Similarity to physical product feature markers (float)*: We adopt here the WordNet Similarity measure [28] to calculate the semantic similarity between a given word and a set of fixed markers of physical product features: "size", "weight", "color", "shape", and "material". The assumption is that the higher the similarity is, the more physical the word is likely to be.
   e. *Plural form (binary)*: In many cases, if a word can be expressed in plural form, then it is more likely to be a physical product feature such as `batteries`, `cases`, etc.

We train SVM classifier with linear kernel and default settings using 250 manually tagged terms and then we evaluate its performance through 5-fold cross validation. Table 2 shows the precision, recall, F_1-score and accuracy of the classifier when applying all the types of features. The results indicate that considering both the semantic meaning and lexical features results in the highest performance in terms of precision (0.874), F_1-score (0.865) and accuracy (0.859). We thus use both semantic and lexical features for extracting physical product features.

Table 2. Evaluation of SVM Classification Model.

Feature selected	Precision	Recall	F_1-score	Accuracy
Semantic + Lexical	**0.874**	0.863	**0.865**	**0.859**
Semantic	0.833	0.812	0.810	0.804
Lexical	0.670	**0.957**	0.788	0.726

6.3 Analyzed Methods

We describe here the baselines and the proposed methods to be tested.

Baselines. We prepare two baselines as follows:

(1) ***Jaccard Coefficient*** (***JaccCoef***): in this method we first detect change periods in the time series of a given candidate effect term. Then, within each change period, we directly compute the Jaccard Coefficient [14] score between any candidate cause term and the effect term. The cause term which has the highest Jaccard Coefficient score will have the highest causality strength. We apply this naïve baseline to examine whether the term co-occurrence would be sufficient to estimate the causality.

(2) ***Lasso Granger Causality*** (***LassoGC***) [2]: this method computes the causality strength based on the theory of Granger Causality [12]. Granger Causality is widely adopted to detect the type-level cause-effect relations. We use it in order to investigate whether the type-level causality detection method could be suitable for our task. ***LassoGC*** is selected as a baseline due to its efficiency and scalability in computing Granger causality within large groups of features. It utilizes Lasso algorithm for linear regression to search for a set of causes C which minimizes the sum of the average squared errors of regressing for an effect e.

Proposed Methods. We test four proposed methods as below.

(1) ***Initial Causality*** (***IniCausal***): this method (see Eq. 4 in Sec. 5.2) is regarded as the basis for the concept of detecting causality between two words over time.

(2) ***Global Causality*** (***GlobCausal***): this method (see Eq. 5 and 6 in Sec. 5.2) is applied to test if considering the global information of all the candidate causes of a given effect can remove spurious causes.

(3) ***Aggregation based on Similar Concept*** (***AggrConcept***): this method (see Sec. 5.3.1) is used for testing if the aggregation by similar concepts helps to generate better results. This aggregation groups both the causes and effects separately and outputs the cause-effect results in a cluster format.

(4) ***Aggregation based on Similar Pattern*** (***AggrPatt***): we apply this method (see Sec. 5.3.2) to test if the aggregation by similar patterns performs differently from the aggregation by similar concepts. In other words, we examine if it is better to aggregate both sides of relations at the same time.

7. EVALUATION

We conduct both quantitative and qualitative evaluation. Their results are described in this section.

7.1 Quantitative Evaluation

7.1.1 Test Sets

As far as we know, no ground truth data is available for the task of token-level causality detection within temporal document collections (i.e., detection of causal relation such that the change in one word implies the change of another word). We have thus manually created test sets containing cause and effect pairs that existed within the time span of each category utilizing external resources including Wikipedia, several dedicated websites [37-48] and a Web search engine. We prepared 54 cause-effect pairs of the category *Electronics, Portable Audio & Video* and 56 pairs for the category *Electronics, Camera & Photo* considering their corresponding time spans. The ground truth data contain two types of effects: actions and usages. Actions are described by verb phrases while use situations are described by nouns such as location terms. Table 6 shows examples of ground truth patterns for the category *Electronics, Portable Audio & Video*. We will consider the output causal pair as correct if both its cause and effect sides are semantically similar to the corresponding sides in any of the ground truth cause-effect pairs. Note that in the ground truth, the effects are sometimes described by verb phrases (e.g., "watch movies"), while the tested methods output either verbs or situation

[6] https://wordnet.princeton.edu/ (accessed on 29/01/2016)

terms as effects. Therefore, when the detected effect is in the form of a verb (e.g., `watch`), we consider it to be correct if its underlying verb matches any verb in the ground truth.

7.1.2 Evaluation Measures

For evaluation, we output up to 10 top results for each year. Then, we combine the results generated for all the years and compare with the ground truth. We compute precision, recall and F_1-score to measure the performance of each method.

Since we make use of two types of time series (frequency-based and semantic-based), we generate the results for each type separately and we evaluate them separately (see columns "Frequency-based" and "Semantic-based" in Tables 3-4). In addition, we also evaluate the performance when combing the results coming from the two types of time series (see column "Freq.-based + Sem.-based" in Tables 3-4). In order to keep the number of returned results the same for different approaches, we merge in each year the top 5 results returned by the method using "Frequency-based" and the one using "Semantic-based" time series.

7.1.3 Evaluation Results

Tables 3-4 describe the performance of each analyzed method. We notice that the proposed methods ***AggrConcept*** and ***AggrPatt*** outperform the two baselines over all the metrics, which proves the proposed approach performs well. We list the other findings below:

(1) ***Co-occurrence is not enough for measuring causality***. According to Tables 3-4, the causality detection is quite difficult as evidenced by quite poor performance of ***JaccCoef***. This suggests that although the co-occurrence describes the relatedness of two terms, it fails to capture the causation (the cause must be a necessary condition for the effect [17]).

(2) ***Time series analysis is not enough for measuring causality within text. LassoGC*** is the typical method for computing the causality between time series. It however delivers poor results when applied for discovering the causality within text. Note that unlike ***LassoGC*** our proposed methods take into consideration both the time series and the probability of term occurrence within text. In contrast to continuous data (e.g., humidity, GDP), where ***LassoGC*** is typically applied, text tends to be more arbitrary and complex. Relying solely on the time series analysis is thus not sufficient for detecting causal patterns in text collections.

(3) ***Computing causality over all candidate causes is necessary***. ***GlobCausal*** has been found to be consistently more effective than ***IniCausal***. This signals that some spurious or weak causal relations are removed by additional filtering that retains genuine causes (see Eqs. 5-6).

(4) ***Aggregation process helps to validate and group cause-effect patterns.*** As we discussed in Sec. 5.3, by aggregating semantically similar binary causal relations, we provide more evidence of the actual causality. For example, the pair `iPod→jogging` is returned at the 68th rank by the method ***GlobCausal***, while ***AggrConcept*** and ***AggrPatt*** return it within the top 5 results. In addition, the grouped similar cause-effect relations have better explanatory power. Another observation is that ***AggrPatt*** performs better than ***AggrConcept*** when using the frequency-based time series.

(5) ***Frequency-based and semantic-based time series complement each other.*** Although methods using both the frequency-based and semantic-based time series are generally effective, their combination helps to discover more correct cause-effect relations than when used alone. This is because there is relatively small overlap between their outputs. Thus, we can say the approaches based on these time series complement each other. This is demonstrated by the improved performance when combing the results from the methods based on each of the time series. The frequency-based time series approaches allow discovering frequent relations. On the other hand, the semantic-based ones help to find the causality between components not commonly mentioned in the dataset, yet, subject to semantic change (i.e., change of the context in which a word is used). This can be observed by analyzing example results in Table 6 (IDs: 18, 23, 26, 33 and 46) which are found by applying methods that utilize the semantic-based time series. The verbs `store`, `share`, `delete`, `surf` and `navigate` are verbs indicating new kinds of actions that became available following the advent of `mp3`, `Napster` and `iPod`.

7.2 Qualitative Evaluation

To further evaluate the quality of the results we also conducted user-based analysis. We invited 5 subjects (2 males and 3 females in their 30s) to annotate the results using several quality criteria.

7.2.1 Settings

Before describing the results, we first clarify the evaluation settings. We utilize the detected physical product features as potential causes, while the extracted verbs and situations are regarded as two types of effects. We also generate two types of time series as described in Sec. 4: the frequency-based and the semantic-based time series. So, in total, we have 4 environment settings considering possible combinations of the effect types and time series types. These are summarized in Table 5.

Table 5. Types of Environment Settings.

Settings	Description
freq_verb	Using frequency-based time series to detect causality between physical features and verbs
freq_sit.	Using frequency-based time series to detect causality between physical features and situation terms
sem_verb	Using semantic-based time series to detect causality between physical features and verbs
sem_sit.	Using semantic-based time series to detect causality between physical feature and situation terms

As discussed above, we have 6 methods to be tested (4 proposed methods and 2 baselines). The cause-effect results by every combination of the method and environment settings are then returned for each of 11 years (the time period when the reviews in the category *Electronics, Portable Audio & Video* of the Amazon Product Review Dataset were created).

The annotators were asked to evaluate the results generated for each year by the 24 approaches (6 methods, each with 4 environment settings). Note that the top 10 results with the highest causality strength are returned on average for each year by every analyzed method. The criteria of the evaluation consist of 3 dimensions: correctness, novelty and comprehensibility (as described in the next section). Each annotator thus gives: 11(years) * 4(environments) * 6(methods) * 3(dimensions) = 792 scorings.

7.2.2 Evaluation Criteria

We used 3 criteria reflecting the general notion of usefulness of the cause-effect relations that ideally should be correct, novel and understandable. The criteria are described below.

Table 3. Results for the Category *Electronics, Portable Audio & Video*.

Method	Frequency-based			Semantic-based			Freq.-based + Sem.-based		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
JaccCoef	0.11	0.18	0.14	0.11	0.18	0.14	0.11	0.18	0.14
LassoGC	0.14	0.22	0.17	0.14	0.22	0.17	0.18	0.28	0.22
IniCausal	0.18	0.28	0.22	0.11	0.18	0.14	0.21	0.34	0.26
GlobCausal	0.20	0.32	0.25	0.19	0.30	0.23	0.29	0.46	0.35
AggrConcept	0.24	0.38	0.30	**0.28**	**0.44**	**0.34**	**0.36**	**0.58**	**0.45**
AggrPatt	**0.28**	**0.44**	**0.34**	0.21	0.34	0.26	0.34	0.54	0.42

Table 4. Results for the Category *Electronics, Camera & Photo*.

Method	Frequency-based			Semantic-based			Freq.-based + Sem.-based		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
JaccCoef	0.10	0.16	0.12	0.10	0.16	0.12	0.11	0.18	0.14
LassoGC	0.16	0.26	0.20	0.23	0.36	0.28	0.26	0.34	0.29
IniCausal	0.13	0.20	0.15	0.21	0.34	0.26	0.30	0.48	0.37
GlobCausal	0.15	0.24	0.18	0.29	0.46	0.35	0.35	0.56	0.43
AggrConcept	0.16	0.26	0.20	0.38	0.60	0.46	0.44	0.70	0.54
AggrPatt	**0.21**	**0.34**	**0.26**	**0.43**	**0.68**	**0.52**	**0.46**	**0.74**	**0.57**

- **Correctness:** it measures how sound the results of each year are. During the assessment, the annotators were allowed to utilize any external resources, such as Wikipedia, Web search engines, books, etc.
- **Novelty:** it measures how novel the results are within the same year. In other words, it quantifies how varying and diverse information the annotators could acquire after viewing the results at a given year.
- **Comprehensibility:** it measures how easy it is to understand and explain the results.

All of the scores were given in the range from 1 to 5 (1: not at all, 2: rather not, 3: so so, 4: rather yes, 5: definitely yes).

7.2.3 Evaluation Results

Correctness. Fig. 5 describes the average correctness scores over each combination of methods and experimental environments. We first notice that the baseline ***LassoGC*** achieves competitive correctness score. It may be because ***LassoGC*** has relatively good performance in detecting frequent and dominant patterns such as `CD player→recording`, `CD player→burn`, `mp3 player→downloading`, etc. Such dominant patterns tend to be highly scored by annotators. As for the evaluation results of other methods, they are basically consistent with the results described in Sec. 7.1.3.

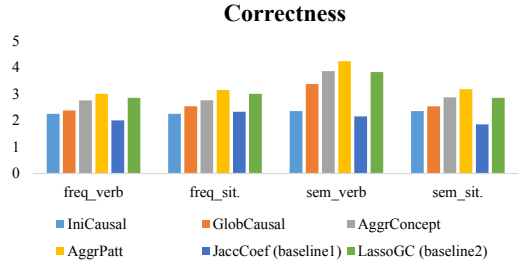


Figure 5. Evaluation of Results based on Correctness.

Novelty. Fig. 6 describes the average novelty scores. The proposed methods based on aggregation procedure, ***AggrPatt*** and ***AggrConcept***, outperform the baselines over each different setting. Even ***GlobCausal*** is better this time than both the baselines when using semantic time series (on average, outperforming ***JaccCoef*** by 57.3% and ***LassoGC*** by 6%). The poor performance of ***LassoGC*** under the novelty criterion implies the limitation of ***LassoGC*** in detecting causality in time series across text collections since this method always gives priority to dominant causes (e.g., `CD`, `mp3`, `iPod`, etc.) or effects (e.g., `download`, `record`, etc.) ignoring infrequent, yet, important ones (e.g., `share`, `delete`, `surf`, `navigate`, etc.). In other words, these results are consistent with the ones shown in Sec. 7.1.3.

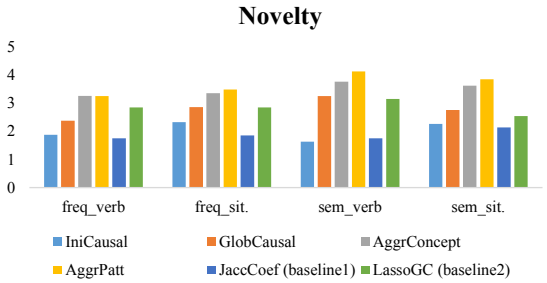


Figure 6. Evaluation of Results based on Novelty.

Comprehensibility. Finally, Fig. 7 compares the comprehensibility scores of the returned results. We note that both the aggregation methods ***AggrConcept*** and ***AggrPatt*** achieve much better performance than the other methods, helping annotators to make more sense from the generated causal relationships.

Table 6. Example results where *Cause* and *Effect* are the ground truth relations. The tags (0, 1) shown in parentheses denote the results using the frequency-based and semantic-based time series, respectively (1 means the results match the ground truth causal relations, while 0 means otherwise).

ID	*Cause→Effect*	JaccCoef (baseline) (*Freq., Sem.*)	LassoGC (baseline) (*Freq., Sem.*)	IniCausal (proposed) (*Freq., Sem.*)	GlobCausal (proposed) (*Freq., Sem.*)	AggrConcept (proposed) (*Freq., Sem.*)	AggrPatt (proposed) (*Freq., Sem.*)
4	radio→car	(0, 0)	(1, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
9	CD player→skip (rewind the track on CD)	(1, 1)	(1, 0)	(1, 1)	(1, 0)	(0, 1)	(0, 0)
10	CD player→recording sound	(1, 1)	(1, 1)	(1, 0)	(1, 0)	(1, 0)	(1, 0)
14	CD player→car	(0, 0)	(0, 1)	(0, 1)	(0, 1)	(1, 1)	(1, 1)
16	CD player→display on panel	(0, 0)	(0, 0)	(0, 0)	(1, 0)	(0, 0)	(1, 0)
18	mp3→store more music	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
23	Napster→share song files	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 0)
26	mp3 player (iPod)→delete songs	(0, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)
30	mp3 player (iPod)→watch movies	(0, 0)	(1, 0)	(1, 0)	(1, 0)	(1, 0)	(1, 0)
32	mp3 player (iPod)→jogging	(0, 0)	(1, 0)	(0, 1)	(0, 1)	(1, 1)	(1, 1)
33	iPod→surf the web	(0, 0)	(0, 0)	(0, 1)	(0, 1)	(0, 1)	(0, 0)
36	mp3 player (iPod)→gym	(0, 0)	(1, 0)	(1, 1)	(1, 1)	(1, 0)	(1, 1)
38	iTunes→download songs	(1, 1)	(1, 1)	(1, 1)	(1, 0)	(1, 1)	(1, 1)
42	iPod→car	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 1)	(0, 0)
46	iPod→navigate song lists	(0, 0)	(0, 1)	(0, 0)	(0, 1)	(0, 1)	(0, 1)

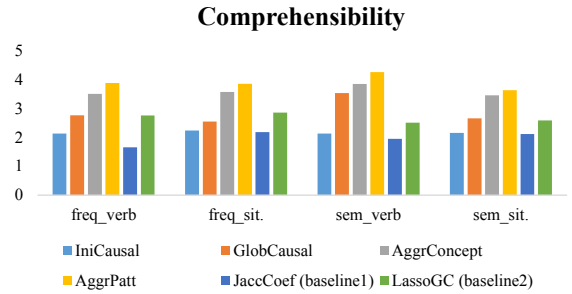


Figure 7. Evaluation of Results based on Comprehensibility

7.2.4 Additional Observations

One interesting finding is that by using our approach, we could also detect certain evolving relationships. In particular, we could track how the technology changed within the same situations. For example, in the results of the category *Electronics, Portable Audio & Video*, we have found that under the situation `car`, the technology changed according to the sequence of `radio>CD>MP3>VCR>TV>iPod`. This suggests that the music devices used in cars changed over time.

Similar type of findings can be also observed in the category *Electronics, Camera & Photo*. By solely looking at the cause side, we can see that the storage media of the camera evolved according to the chain of {`film`, `tape`}`>sd card>memory card>DVD`. We can also observe different actions involved with the new storage media. For example, at the very beginning, people utilized {`Kodak`, `film`} to take a photo and they needed to `print` the photos, while, in the later years, cameras became more advanced utilizing digital formats of data that can be transferred externally such as via `sd card`. The corresponding actions changed then to {`plug`, `transfer`, `convert`} indicating new possibilities for viewing photos such as by using computers.

8. CONCLUSIONS

Technology and product evolution is an interesting, yet, at the same time, quite challenging concept for analysis. Given the abundance of product review collections that already span relatively long time periods we propose to utilize them for automatically detecting knowledge about technology progress over time. The intriguing characteristics of our research is a novel approach in which we attempt at automatically estimating the effects of technology and product evolution on our lives. For this we specifically propose detecting cause-effect relations on word time series. We use not only the frequency-based time series but also propose constructing time series that capture changes in word usage over time by applying neural networks. Both temporal representations provide complementary results as demonstrated in the experiments. Furthermore, we aggregate the binary causal relations for obtaining correct and comprehensible causal patterns. Experimental evaluation demonstrates that our methods outperform baselines both when compared with ground truth as well as when evaluated by annotators.

We believe that the proposed methods can be also applied to other scenarios besides the product and technology evolution (e.g., in the collections of news articles or scientific publications) with minor adaptations. For example, in this paper, we decrease the number of candidate term pairs for causality estimation by constraining the terms to those involving the cause as a technology feature (candidate causes) and to those related to way in which technology is used (candidate effects). Yet, different constrains on selecting the candidate causes and effects can be applied in other domains (e.g., named entities in news article collections or scientific terminology in scientific publication collections).

9. ACKNOWLEDGMENTS

This work was supported in part by Grants-in-Aid for Scientific Research (Nos. 15H01718, 15K12158) from MEXT of Japan and by the JST Research Promotion Program Presto (Sakigake): “Analyzing Collective Memory and Developing Methods for Knowledge Extraction from Historical Documents”.

10. REFERENCES

[1] J. Allan. Topic Detection and Tracking: Event-based Information Organization, *Science & Business Media*, vol. 12, Springer, 2002.

[2] A. Arnold, Y. Liu, N. Abe. Temporal Causal Modeling with Graphical Granger Methods. In *Proc. of SIGKDD 2007*, pp. 66-75, 2007.

[3] K. Berberich, S. J. Bedathur, M. Sozio and G. Weikum, Bridging the Terminology Gap in Web Archive Search, In *Proc. of WebDB'09*, 2009.

[4] W. E. Bijker, P. H. Thomas, and T. J. Pinch, *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. Cambridge, MA: MIT Press, 1987.

[5] E. Billauer. PEAKDET: http://billauer.co.il/peakdet.html (accessed on 29/01/2016)

[6] D. Blei and J. Lafferty. Dynamic Topic Models, In *Proc. of ICML 2006*, pp. 113-120, 2006.

[7] G. A. Cohen, *Karl Marx's Theory of History: a Defense*. Oxford University Press, 2000.

[8] G. F. Cooper, E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine learning*, 1992, 9(4): 309-347.

[9] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. J. Gao et al., TextFlow: Towards Better Understanding of Evolving Topics in Text, *IEEE Transaction on Visualization and Computer Graphics*, vol. 17, No. 12, 2011.

[10] P. D. Gilbert. Combining VAR Estimation and State Space Model Reduction for Simple Good Predictions. *Journal of Forecasting*, 1995, 14(3): 229-250.

[11] R. Girju, D.I Moldovan. Text Mining for Causal Relations. In *Proc. of FLAIRS 2002*, pp. 360-364, 2002

[12] C. W. J. Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica: Journal of the Econometric Society*, 37 (3): 424–438, 1969.

[13] H. Hansson, B. Jonsson. A Logic for Reasoning about Time and Reliability. *Formal aspects of computing*, 1994, 6(5): 512-535.

[14] Jaccard, Paul, Étude Comparative de la Distribution Florale dans une Portion des Alpes et des Jura, *Bulletin de la Société Vaudoise des Sciences Naturelles* 37: 547–579, 1901.

[15] A. Kalurachchi, A. S. Varde, S. Bedathur, G. Weikum, J. Peng and A. Feldman, Incorporating Terminology Evolution for Query Translation in Text Retrieval with Association Rules, In *Proc. of CIKM 2010*, pp. 1789-1792, 2010.

[16] N. Kanhabua, K. Nørvåg, Exploiting Time-based Synonyms in Searching Document Archives, In *Proc. of JCDL2010*, pp. 79-88

[17] S. Kleinberg. *Causality, Probability, and Time*. Cambridge University Press, 2012.

[18] S. Kleinberg, G. Hripcsak. A Review of Causal Inference for Biomedical Informatics. *Journal of Biomedical Informatics*, 2011, 44(6): 1102-1112.

[19] S. Kleinberg, B. Mishra. The Temporal Logic of Causal Structures. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence* (UAI-09), pp. 303-312, 2009.

[20] V. Kulkarni , R. Al-Rfou, B. Perozzi, et al. Statistically Significant Detection of Linguistic Change, In *Proc. of WWW 2014*, pp. 625-635, 2014.

[21] A. Mazeika, T. Tylenda and G. Weikum. Entity Timelines: Visual Analytics and Named Entity Evolution, In *Proc. of CIKM 2011*, pp. 2585-2588, 2011.

[22] J. McAuley and J. Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proc. of RecSys 2013*, pp. 165-172, 2013.

[23] R. Mihalcea, P. Tarau, TextRank: Bringing Order into Texts. In *Proc. of ACL 2004*, 2004.

[24] T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *Proc. of ICLR*, 2013.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representation of Phrases and Their Compositionality. In *Proc. of NIPS 2013*, pp. 3111-3119, 2013.

[26] P. Mirza. Extracting Temporal and Causal Relations between Events. In *Proc. of ACL 2014 Student Research Workshop*, pp. 10-17, 2014.

[27] S. Payson. Product Evolution: What it is and How it Can be Measured. *Eastern Economic Journal*, 1995: 247-262.

[28] T. Pedersen, S. Patwardhan, J. Michelizzi, WordNet: Similarity: Measuring the Relatedness of Concepts, In *Proc. of HLT-NAACL 2004*, pp. 38-41, 2004.

[29] P. Radhakrishnan, M. Gupta, V. Varma. Modeling the Evolution of Product Entities. In *Proc. of SIGIR 2014*, pp. 923-926, 2014.

[30] K. Radinsky, S. Davidovich, S. Markovitch. Learning Causality for News Events Prediction. In *Proc. of WWW 2012*, pp. 909-918, 2012.

[31] D. E. Rumelhart, G. E. Hinton, R.J. Williams. Learning Internal Representations by Error Propagation. California Univ, San Diego La Jolla Inst. For Cognitive Science, 1985.

[32] P. Spirtes, C.N. Glymour, R. Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

[33] J. Strötgen, O. Alonso, M. Gertz. Retro: Time-based Exploration of Product Reviews, *Advances in Information Retrieval. Springer Berlin Heidelberg*, pp. 581-582, 2012.

[34] N. Tahmasebi, G. Gossen, N. Kanhabua, H. Holzmann, and T. Risse. NEER: An Unsupervised Method for Named Entity Evolution Recognition, In *Proc. of Coling 2012*, pp. 2553-2568, 2012.

[35] Y. Zhang, A. Jatowt, K. Tanaka. Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time. In *Proc. of ACL 2015*, pp. 645-655, 2015.

[36] J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary Hierarchical Dirichlet Process for Multiple Correlated Time-varying Corpora, In *Proc. of SIGKDD*, pp. 1097-1088, 2010.

[37] A Complete History of Portable Music Players, 2014, http://www.ebay.com/gds/A-Complete-History-of-Portable-

Music-Players-/10000000177628958/g.html (accessed on 29/01/2016)

[38] The History of Portable Audio, 2001, http://www.ehow.com/about_5292437_history-portable-audio.html (accessed on 29/01/2016)

[39] Thank you for the Music: A Potted Pictorial History of Portable Music Devices, 2014, http://www.telegraph.co.uk/news/picturegalleries/uknews/10938261/Thank-you-for-the-music-A-potted-pictorial-history-of-portable-music-devices.html (accessed on 29/01/2016)

[40] Top 10 Historical Music Players, 2013, http://cassette-to-mp3-review.toptenreviews.com/top-10-historical-music-players.htmls (accessed on 29/01/2016)

[41] The History of Car Radios, 2010, http://www.caranddriver.com/features/the-history-of-car-radios (accessed on 29/01/2016)

[42] Top Five MP3 Players for Running, 2015, http://aminebombom.hubpages.com/hub/top-5-mp3-players-for-running (accessed on 29/01/2016)

[43] The Disadvantages of Film Cameras, 2012, http://www.ehow.com/info_8078035_disadvantages-film-cameras.html (accessed on 29/01/2016)

[44] Camera Innovation: 10 Products that are Changing how We Take Photos and Videos, 2014, http://www.techrepublic.com/article/camera-innovation-10-products-that-are-changing-how-we-take-photos-and-videos/ (accessed on 29/01/2016)

[45] Digital Camera Advantages, 2009, http://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/knowhow25.html (accessed on 29/01/2016)

[46] External Flash, 2008, http://photographycourse.net/lessons/external-flash/ (accessed on 29/01/2016)

[47] 10 Things You Need to Know About Camera Lenses , 2014, http://www.ebay.com/gds/10-Things-You-Need-to-Know-About-Camera-Lenses-/10000000177628167/g.html (accessed on 29/01/2016)

[48] Benefits and Limitations of Dslrs Vs. Camcorders, 2012, http://www.videomaker.com/videonews/2012/07/benefits-and-limitations-of-dslrs-vs-camcorders (accessed on 29/01/2016)