# Learning Causality from Textual Data

**Kira Radinsky*, Sagie Davidovich† and Shaul Markovitch***
Computer Science Department
Technion–Israel Institute of Technology
*{kirar, shaulm}@cs.technion.ac.il, †{mesagie}@gmail.com

## Abstract

We present a new methodology for modeling and predicting future events through machine learning and data mining techniques from textual data. Modeled events span across varied domains including politics, economy and society. The model employs human-style prediction techniques such as causality inference, generalization and projection based on past experience. For this purpose, we use news archives that date back 150 years as a vast source of text representing "past-experiences" and inference patterns. Empirical evaluation on real news articles shows that the ability of our algorithm to predict future events is similar to that of humans.

## 1 Introduction

Predicting events in politics, economics, society, etc. is an intriguing task that is usually performed by human experts possessing extensive domain-specific and common-sense knowledge. Much of the causal knowledge that helps humans understand the world is found in texts, that expresses people's beliefs. Understanding causality and the ability to predict are fundamental capabilities of intelligent behavior and are essential for decision making and other human common-sense reasoning, such as question-answering.

Psychological studies [Kahneman and Tversky, 1973] provide evidence that humans are good at certain types of event predictions – especially as they base on their lifetime knowledge about how the world behaves. They posses wide common-sense knowledge about how the world is now, how it was before, what happens after an action, and which event causes another.

Can we endow a machine with such rich capabilities to allow it to predict events? Specifically, can we obtain all this expertise which humans are exposed to throughout their lifetime experience? The World Wide Web encapsulates much of our humans knowledge. It treasures information about historical events through news archives and encyclopedias. It has dynamic updates about current events through online news papers updating every minute. On top of all the past and current human history, the web has many common sense ontologies that can be utilized to extract causality patterns and generalize events. This knowledge can serve as the basis for performing true human-like prediction – with the ability to learn, understand language, possess intuitions and general world knowledge.

Although some works dealt with entity and relation extraction [Carlson *et al.*, 2010], including causal extraction specifically [Chan and Lam, 2005], and temporal information extraction [Ling and Weld, 2010] from the web, to the best of our knowledge, none have dealt with causality prediction. Other related works deal with information overflow – a somehow related work is [Shahaf and Guestrin, 2010] which, given two news articles, provides a coherent small number of news items that connects them. In our work we receive an event and predict its effect.

Time and time again we can draw parallels from what is happening today, to historical examples in the past. And although it is impossible for history to repeat itself exactly, there are indeed recurring themes and patterns, in the historical time line of mankind. Therefore, finding similar occurrences in the past, and observing what they caused, might give us great insight on the present. Interestingly, psychological studies ([Kahneman and Tversky, 1973]) show that people also predict by similarity. We present a new methodology for predicting events using a learning algorithm, which given an event represented in natural language, predicts a future event it can cause. We first introduce a method for event representation inspired by Kim's (1993) property exemplification of events theory and Schank's (1972) conceptual dependency theory. We then present a prediction algorithm, that uses world knowledge to generalize the events it was trained on in the past. During prediction, when presented with a never-seen-before event, it matches the event to a similar generalized or specific event in the past. It then projects the present event to a predicted future event, using a cause-effect clause pattern it learnt during training. For example, given a present event "elections in Tehran", and a matched past causality of "elections in Baghdad" caused "protests in Iraq", the algorithm will output "protests in Iran" rather than the naive output

of "protests in Iraq".

For training the algorithm, we have created a graph of 300 millions fact nodes connected by more than one billion edges. Temporal data expressing causality was obtained mainly from the New-York-Times archives (dating back to 1851). We applied semantic natural language modeling techniques on the textual data, creating a structured representation of the knowledge.

We present an empirical framework for evaluating the performance of the algorithms. Given a set of events extracted from the 2010 news, humans were asked to indicate what they believe would happen as a result of these events. Given the human predictions and the algorithm predictions of those events, we asked an additional group to conclude which predictions seemed more logical. The results suggest that the ability of the algorithm to predict future events is at least as good as the human ability to predict.

To gain some intuition about the type of predictions the algorithm issues we present here two examples. The algorithm, given the event "Magnitude 6.5 earthquake rocks the Solomon Islands", predicted that "Tsunami-warning will be issued in the Pacific Ocean". It learnt this based on past examples it was trained on, one of which was "Tsunami warning issued for Indian-Ocean" after "7.6 earthquake strikes island near India". The predictive template inferred by the algorithm was: if an earthquake occurs next to an Island, a tsunami warning will be issued for its nearest ocean. An additional example of a prediction issued by the algorithm, is given the event "Cocaine found at Kennedy Space Center", it outputted the following predictions: "few people will be arrested", as the past event "police found cocaine in lab" caused the event "2 people arrested". As can be seen from the examples, not all of the predicted events will really take place in reality. However, most people will agree that they do seem logical.

The main contributions of this paper are threefold: First, we present a method of modeling events and construct the world's largest causality graph, using novel causality mining techniques and data sources. Second, we present a novel method for prediction of general future events using their patterns in the past. Finally, we present a new architecture design for mining events, and a testing methodology for evaluating news prediction algorithms, and evaluate our algorithms.

# 2 Problem Definition

In this section we discuss the event-causality inference problem.

## 2.1 Event Representation

One of the theories to discuss how an event should be represented is Kim's Property Exemplification of Events theory [Kim, 1993]. Kim's discussion starts with the assumption of the availability of a set of entities $O$. These entities represent physical and abstract objects in the real world: people, instances, and types. Our representation of events complies with this theory, which, in broad strokes, states that events are structured and defined by a triplet $[O, P, t]$: (1) An object or several objects ($O_i \subseteq O$); (2) a relation or property ($P \subseteq O \times O$), and (3) a time interval ($t$).

Kim's theory provides a hight-level view on events. To enhance the structure representation of an event we turn to Conceptual Dependency (CD) [Schank, 1972] theory. Inspired by this this theory, we further structure the event to have roles in addition to the property relation. Each event will be composed of a temporal action or state that the event's objects exhibit ($P$), the *actor* that conducted the action, the *object* on which the action was performed, the *instrument* the action was conducted with, and the *location* of the event. Formally, it is represented as an ordered set $e = \langle P, O_1, \ldots, O_4, t\rangle$, where $P$ is the action, $O_i \subseteq 2^O$ and $t$ is a time-stamp . For example, the event "The U.S army destroyed a warehouse in Iraq with explosives", which occurred on October 2004, is modeled as: Destroy (Action); U.S Army (Actor); warehouse (Object); explosives (Instrument); Iraq (Location); October 2004(Time).

## 2.2 Learning Problem Definition

We treat causality inference as a learning problem. Let $Ev$ be the set of events as described above. The goal concept is a set of ordered-pairs of events that share a causality relation among them: $Gc = \{\langle e_i, e_j\rangle | e_i, e_j \in Ev\}$, where $e_i$ is an event causing $e_j$. Given training examples $T \subseteq Gc$, the algorithm produces a predictor $h : Ev \rightarrow Ev$ based on these examples. The predictor can then be applied on a new event $e_i \in Ev$ to output an element in the goal concept $\langle e_i, e_j\rangle \in Gc$.

## 2.3 Generalizations

The goal of learning is to generalize the training examples supplied, in order to be able to handle never seen new examples. The learning examples are composed of events, which in turn are composed of two main components – the objects and the properties that they hold at the time of the event. In this section we present how both components of the event can be generalized.

### Generalizing Objects

A common practice is to map an object $o \in O$ to a concept $c \in C$. To allow generalization of the objects we assume the availability of a concept graph over those concepts. Formally, we assume a concept graph $G_O = (V, E)$, where $V \subseteq 2^O$. We also assume a labeling on the edges that represents the relation between the objects.

### Generalizing Properties

In order to generalize the actions in the events we adopt the Conceptual Dependency (CD) paradigm of actions groups [Schank, 1972]. In this theory, Schank discusses how to map each action to 11 classes of actions, such as: move, PTrans, speak, Mbuild etc. This enables us to classify the event property $P$ to higher action classes.

### 2.4 Hypothesis Space

We define an event $Gen(e) = e' = \langle P', O'_1, \ldots, O'_4, t\rangle$ to be a generalization of an event $e = \langle P, O_1, \ldots, O_4, t\rangle$, if there exists a path in the graph $G_o$ between $O_i$ to $O'_i$ and if $P'$ and $P$ belong to the same class of actions. The hypothesis space is the set of all possible predictors $h : Ev \rightarrow Ev$. The hypothesis in our learning scheme is a generalized pair of events: $\langle Gen(e_i), Gen(e_j)\rangle$, $e_i \in Ev$.

## 3 The Prediction Algorithm

In this section we present a learning algorithm that learns how to generalize from past events in order to produce a predictor $h$, that given a present event can predict its effect. The solution builds on the case-based reasoning (CBR) framework [Aamodt and Plaza, 1994]. Given training examples and a new instance, this framework is described by 4 main stages: retrieving stored training examples most similar to the new instance (*retrieving*), combining the matched training instance with the new instance to produce a new solution(*reusing*), updating the stored instances (*revising*), and finally adding the new instance to the stored examples (*retaining*). In this work we implement the first two stages.

**Retrieving Stage** In order to define the similarity of two events, we first define the *Events Edit Distance* $d_e$. Let $e_i = \langle P^i, O^i_1, \ldots, O^i_4, t^i\rangle$ and $e_j = \langle P^j, O^j_1, \ldots, O^j_4, t^j\rangle$ be two events. Let $G^u_o$ and $G^u_p$ be the undirected versions of the objects graph $G_o$ and the action graph $G_p$ respectively [1]. We define $d_e(e_i, e_j) = dist_{G^u_p}(P^i, P^j) + \sum_k dist_{G^u_o}(O^i_k, O^j_k)$, where, $dist_G$ is the length of the shortest path among all paths in $G$.

Given a new event $e_i$, the algorithm retrieves the top most similar events $S = \{\langle e, e_m\rangle \in T\}$ by $d_e$. Those matched events are then generalized: $Gen(S) = \{\langle e_1, e_2\rangle | \exists e, e_m \in Ev : \langle e, e_m\rangle \in S, e_1 \in Gen(e), e_2 \in Gen(e_m)\}$. We define *specificity weight* of an event to be $\xi(e) = |\{\langle e, e_j\rangle \in S \bigcup Gen(S) | e, e_j \in Ev\}|$, representing the variance in the effects of the event $e$, and a *support score* $\vartheta(\langle e'_i, e'_j\rangle) = |\{\langle e_i, e_j\rangle \in S | e'_i \in Gen(e_i), e'_j \in Gen(e_j)\}$, which indicates how many supportive evidence in the examples we have for this event pair or generalized event pair. The final similarity measurement of two events is: $Sim(e_i, e_j) = \frac{max_{e_m, \langle e_i, e_m\rangle \in Gen(S)} \vartheta(e_i, e_m) \cdot d_e(e_i, e_j)}{\xi(e_i)}$. In principal, we prefer matched events or generalized events which had many examples in the training (support) and cause little specific events (i.e. were not generalized too much). We apply the described distance on the cause events in $S \bigcup Gen(S)$, retrieving the top most similar events.

**Reusing Stage** Predicting the effects of the matched events directly has some drawbacks. Assume an event $e_i$="Earthquake hits Haiti" occurred today, and during retrieving, it was matched to the pair ⟨"Earthquake hits Turkey" , "Red Cross help sent to Ankara"⟩. Obviously, predicting that Red Cross help will be sent to Ankara because of an earthquake in Haiti is not logical. We would like to be able to generalize the past cause and effect pair and learn a predicate clause that connects them, e.g. for "Earthquake hits [Country Name]" yield "Red Cross help sent to [Capital of Country]". During the reusing stage, such a clause will be applied to the present event $e_i$ producing its effect with regard to $e_i$. In our example, the logical predicate clause would be *CapitalOf*, as *CapitalOf(Turkey)*= Ankara. When applied on the current event $e_i$: *CapitalOf(Haiti)* = Port-au-Prince, the output will now be "Red Cross help sent to Port-au-Prince". Notice that the application of the clauses can only be applied on certain types of objects – in our case, countries. The clauses can be of any length, e.g., ⟨"suspect arrested in Brooklyn", "Bloomberg declares emergency"⟩ produces the clause *Mayor(BoroughOf(x))*, as Brooklyn is a borough of New York, whose mayor is Bloomberg.

We will now show how to learn such clauses, and how they should be applied. Recall that the graph $G_O$ is an edge-labeled graph, where each edge is a triplet $\langle v_1, v_2, l\rangle$, where $l$ is a predicate (e.g. "CapitalOf"). The learning procedure is divided to 3 main steps: First, finding an undirected path $p_i$ of length at most $k$ in $G_O$ between the objects of the cause event to the effect event; Second, constructing a clause using the labels of the path $p_i$ as the predicates. We call this the *predicate projection* of size $k$ , $pred = l_1, \ldots, l_k$ from an event $e_i$ to an event $e_j$. Finally, the projection is applied to the new event $e = \langle P^i, O_1, \ldots, O_4, t\rangle$ by finding an undirected path in $G_O$ from $O_i$ with the labels of *pred*. The projection results are all the objects in the vertex reached. Formally, *pred* can be applied if $\exists V_0 : O \subseteq V_0, \exists V_1 \ldots V_k : (V_0, V_1, l_1), \ldots (V_{k-1}, V_k, l_k) \in Edges(G^u_O)$. The projection results are all the objects $o \in V_k$.

## 4 Mining Causality

In the previous section we present a high-level algorithm that requires training examples $T$, knowledge about entities $G_O$, and events' action classes $P$. One of the main challenges of this work was to build a scalable system to obtain those requirements.

We present a system that mines news sources to extract events, constructs their canonical semantic model, and builds a causality graph on top of those events. The system crawled, for more than 4 months, several dynamic information sources, the main one being the New-York-Times archives (which on part of the data optical character recognition (OCR) was performed), gathering data of more than 150 years ($1851 - 2009$).

For generalization of the objects, the system automatically reads web content and extracts world knowledge. The knowledge was mined from structured and semi-structured publicly available information repository. The causality graph building was distributed over 20 machines, using a Map-Reduce framework. This process efficiently unites different sources, extracts events,

[1] The actions are grouped into classes, but we treat it as a 2-level graph.

and disambiguates entities. The resulting causality graph is composed of over 300 million entity nodes, one billion static edges and over 7 million causality edges.

On top of the causality graph, a search and indexing infrastructure was built to enable search over millions of documents. This highly scalable index allows a fast walk on the graph of events, enabling efficient inference capabilities during the reusing phase of the algorithm.

### 4.1 World knowledge mining

The entity graph $G_o$ is composed of concepts from Wikipedia, ConceptNet[Liu and Singh, 2004], WordNet[Miller, 1995], Yago[Suchanek *et al.*, 2007], and OpenCyc. The concepts are interlinked using Linked-Data cloud (e.g., DBpedia). The billion labeled edges of the graph $G_o$ are the predicates of those ontologies.

### 4.2 Causality events mining and extraction

Our supervised learning algorithm requires many learning examples to be able to generalize well. As the amount of temporal data is extremely large, spanning over millions of articles, the goal of getting human annotated examples becomes impossible. We therefore provide an automatic procedure to extract labeled examples for learning causality from dynamic content. Specifically in this work, we used the New-York-Times archives for the years $1851 - 2009$, WikiNews and BBC – over 13 million articles in total.

The system mines unstructured natural language text, found in the dynamic web content, and searches for causal grammatical patterns. We construct those patterns using *causality connectors* [Wolff *et al.*, 2002; Levin and Hovav, 1994]. The connectors are divided to three groups: causal connectives (e.g. because, after), causal prepositions (e.g. due to, because of) and periphrastic causative verbs (e.g. cause, lead to). We constructed a set of rules for extracting a causality pair. Each rule is structured as: ⟨Pattern, Constraint, Priority⟩, where Pattern is a regular expression containing a causality connector, Constraint is a syntactic constraint on the sentence on which the pattern can be applied, and Priority is the priority of the rule if several rules can be matched. For example, for the causality connector "after", the pattern "After [sentence1], [sentence2]" is used, with the constraint that [sentence1] cannot start with a number. This pattern can match the sentence "after Afghan vote, complaints of fraud surface" but will not match the sentence "after 10 years in lansing, state lawmaker Tom George returns". An additional pattern example is "[sentence1] as [sentence2]" with the constraint of [sentence2] having a verb. Using the constraint, the pattern can match the sentence "Nokia to cut jobs as it tries to catch up to rivals" is matched, but not the sentence "civil rights photographer unmasked as informer". The result of a rule application is a pair of sentences – one tagged as a cause, and one tagged as an effect.

Given an extracted natural-language sentence, representing an event (either during learning or prediction), the following procedure transforms it into a structured event:

1. Root forms of inflected words are extracted using a morphological analyzer derived from WordNet [Miller, 1995] stemmer. For example, in the article title from 10/02/2010: "U.S. attacks kill 17 militants in Pakistan", the words "attacks", "killed" and "militants" are transformed to "attack", "kill" and "militant" respectively.
2. Part-Of-Speech tagging ([Marneffe *et al.*, 2006]) is performed, and the verb is identified. The class of the verb is identified using the VerbNet vocabulary [Hoa Trang Dang and Rosenzweig, 1998], e.g., kill belongs to $P$ =murder class.
3. A syntactic template matching the verb is applied to extract the semantic relations and thus the roles of the words. Those templates are based on VerbNet, which supplies for each verb class a set of syntactic templates. These templates match the syntax to the thematic roles of the entities in the sentence. We match the templates even if they are not continuous in the sentence tree. This allows the match of a sentence even where there is an auxiliary verb between the subject and the main transitive verb. In our example, the template is "NP1 V NP2" which transforms NP1 to "Agent", and NP2 to "Patient". Therefore, we match U.S. attacks to be the *Actor*, and the militant to be the *Patient* . If no template can be matched, the sentence is transformed into a typed-dependency graph of grammatical relations [Marneffe *et al.*, 2006]. In the example, U.S. attacks is identified as the subject of the sentence (candidate for Actor), militants as the object (candidate for Patient), and Pakistan as the preposition (candidate for Location or Instrument, based on heuristics, e.g., locations lexicons). Using this analysis, we identify that the *Location* is Pakistan.
4. Each word in $O_i$ is mapped to a Wikipedia-based concept. If a word matches more than one concept, we perform disambiguation by computing the cosine similarity between the body of the news article and the body of the Wikipedia article associated with the concept: e.g., U.S was matched to several concepts, such as: United States, University of Salford, and Us (Brother Ali album). The most similar by content was United States Wikipedia concept.
5. The time of the event $t$ is the time of the publication of the article in the news, e.g., $t$ =10/02/2010.

In our example, the final result is the event $e$ = ⟨MurderClass, United States Of America, Militant, NULL ,Pakistan, 10/02/2010⟩ .

In many cases additional heuristics were needed in order to deal with the briefness in news language, e.g:

1. Missing Context – In "McDonald's recalls glasses due to Cadmium traces", the extracted event "Cadmium traces" needs additional context – "Cadmium traces [in McDonald's glasses]". Heuristically, if an

object is missing, the first sentence ([sentence1]) subject is used.

2. Missing entities and verbs – the text "22 dead" should be structured to the event "22 [people] [are] dead". Heuristically, if a number appears as the subject, the word people is added and used as the subject, and "be" is added as the verb.
3. Anaphora resolution – the text "boy hangs himself after he sees reports of Hussein's execution" is modeled as "[$boy_1$] sees reports of Hussein's execution" causes "[$boy_1$] hangs [$boy_1$]" [Lappin and Leass, 1994].
4. Negation – the text "Matsui is still playing despite his struggles" should be modeled as: "[Matsui] struggles" causes the event "Matsui is [not] playing". Modeling preventive connectors (e.g., despite) requires negation of the modeled event.

# 5 Empirical Evaluation

A variety of experiments were conducted to test the performance and behavior of our algorithm.

## 5.1 Methodology

We implemented the algorithms described above and evaluated their performance. The prediction algorithm was trained using news articles from the period 1851 – 2009. The web resources snapshots mentioned in Section 4 dated until 2009. The evaluation is performed on separate data – Wikinews articles from the year 2010. We refer to this data as the *test data*. The evaluation procedure is divided to the following steps:

1. Event identification – our algorithm assumes that the input to the predictor $h$ is an event. To find news titles that represent an event, we randomly sample $n$ headlines from the test data. For each headline a human is requested to decide whether the headline is an event which can cause other events. We denote the set of headlines labeled as event as $E$. We again randomly sample $k$ titles from $E$. We denote this group as $C$.
2. Algorithm event prediction – we run our algorithm on each event title $c_i \in C$. The algorithm performs event extraction from the headline, and produces an event $e_i^a$ with the highest score of being caused by the event represented by $c_i$. The result of this stage are the pairs: $\{(c_i, e_i^a) | c_i \in C\}$.
3. Human event prediction – we present a human with an event title $c_i \in C$, asking what this event might cause. We denote the human result as $e_i^h$. The human is requested to provide the answer in a structured manner (as our algorithm produces). The result of this stage are the pairs: $\{(c_i, e_i^h) | c_i \in C\}$.
4. Human evaluation of the results – Present $m$ people with a triplet $(c_i, e_i^h, e_i^a)$. We ask to evaluate the precision of the pair: $(c_i, e_i^h)$ and the precision of $(c_i, e_i^a)$, on a scale of 0-4 (0 is a highly impossible prediction and 4 is a highly possible prediction).

| | [0-1) | [1-2) | [2-3) | [3-4] | Average Ranking | Average Accuracy |
|---|---|---|---|---|---|---|
| Algorithm | 0 | 2 | 19 | 29 | 3.08 | **77%** |
| Humans | 0 | 3 | 24 | 23 | 2.86 | **72%** |

Table 1: The histogram of the rankings of the users for both human and algorithm predictions.

Human evaluation was conducted using Amazon Mechanical Turk, an emerging utility for performing user study evaluations, which was shown to be very precise for certain tasks [Kittur *et al.*, 2008]. During the evaluation, tasks are created by routing a question to random users and obtaining their answers.

## 5.2 Results

We performed the above mentioned experiments, with the values $n = 1500, k = 50, m = 10$. The algorithm average prediction precision was 3.08/4 (3 is a "possible prediction"), and the human prediction average precision was 2.86/4. For each event, we average the results of the $m$ rankers, producing an averaged score for the algorithm performance on the event, and an averaged score for the human prediction (see Table 1). We performed a paired t-test on the $k$ paired scores. The advantage of the algorithm over the humans was found to be statistically significant with $p < 0.05$. We can conclude now that the ability of the algorithm to predict future events is at least as good as the human ability to predict.

We now present qualitative analysis of the results to have a better understanding of the algorithm strength and weaknesses. Given the event "Louisiana flood" the algorithm predicted that [number] people will flee. The prediction was based on the following past news articles: Residents of Florida flee storm and Hiltons; 150000 flee as hurricane nears north Carolina coast; a million flee as huge storm hits Texas coast; Thousands in Texas flee hurricane Ike; thousands flee as storm whips coast of Florida; at least 1000 flee flooding in Florida. The past events were generalized to the causality pair of "[Weather hazards] at [States of the Southern United States]" cause "[number] of people to flee". During the prediction, the event "Louisiana flood" (which did not occur in the training examples) was found most similar to the above generalized causality pair.

As another example, given the event "6.1 magnitude aftershock earthquake hits Haiti", it outputted the following predictions: "[number] people will be dead", "[number] people will be missing", "[number] magnitude aftershock earthquake will strike island near Haiti" and "earthquake will turn to United States Virgin Islands". While the first 3 predictions seem very reasonable, the fourth one is problematic. The rule the system learnt in this case is – natural disasters that hit countries next to a shore tend to affect near by countries. In our case it predicted that the earthquake will affect United States Virgin Islands, which are geographically close to Haiti. However, the prediction "earthquake will turn to United States Virgin Islands" is not very realistic as an earth-

| Event | Human-predicted event | Algorithm-predicted event |
|---|---|---|
| Al-Qaida demands hostage exchange | Al-Qaida exchanges hostage | A country will refuse the demand |
| Volcano erupts in Democratic Republic of Congo | Scientists in Republic of Congo investigate lava beds | Thousands of people flee from Congo |
| 7.0 magnitude earthquake strikes Haitian coast | Tsunami in Haiti affects coast | Tsunami-warning is issued |
| 2 Palestinians reportedly shot dead by Israeli troops | Israeli citizens protest against Palestinian leaders | War will be waged |
| Professor of Tehran University killed in bombing | Tehran students remember slain professor in memorial service | Professor funeral will be held |
| Alleged drug kingpin arrested in Mexico | Mafia kills people with guns in town | Kingpin will be sent to prison |
| UK bans Islamist group | Islamist group would adopt another name in the UK | Group will grow |
| China overtakes Germany as world's biggest exporter | German officials suspend tariffs | Wheat price will fall |

Table 2: Human and algorithm predictions for events.

quake cannot change its course. It was created based on a match with a past example of a tornado hitting a country on a coast. The reason for that is the sparsity of the training. Both are natural disasters, and there were no negative examples or enough positive examples to support this distinction. However, we still find this example interesting, as it issues a prediction using spatial locality (United States Virgin Islands are [near] Haiti). Another example of the same problem is the prediction: ⟨ lightning kills 5 people, lightening will be arrested⟩, which was predicted based on training examples in which people who killed other people got arrested.

Additional 8 examples out of the 50 in the test and their predictions can be seen in Table 2.

## 6 Conclusions

We presented a method for representing events based on previous philosophical contribution. We introduced a novel case-based reasoning algorithm for predicting causality relations between events. Our prediction algorithm uses an extensive knowledge base which was automatically constructed using information mined from large amount of text. We presented the data mining and natural language techniques to transform the raw data of over 150 years of history archives into a structured representation of events, using a mined web-based object hierarchy and action classes. Our experimental evaluation showed that the predictions of the algorithm are at least as good as those of humans.

We believe that our work is one of the first to harness the vast amount of information available on the web to perform prediction that is general purpose, knowledge based, and human like.

## References


[Aamodt and Plaza, 1994] Agnar Aamodt and Enric Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.*, 7:39–59, 1994.

[Carlson *et al.*, 2010] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. Toward an architecture for never-ending language learning. In *Proc. AAAI*, 2010.

[Chan and Lam, 2005] Ki Chan and Wai Lam. Extracting causation knowledge from natural language texts. *IJIC*, 20:327–358, 2005.

[Hoa Trang Dang and Rosenzweig, 1998] Martha Palmer Hoa Trang Dang, Karin Kipper and Joseph Rosenzweig. Investigating regular sense extensions based on intersective levin classes. In *Proc. Coling-ACL*, 1998.

[Kahneman and Tversky, 1973] D. Kahneman and A. Tversky. On the psychology of prediction. *Psychological review*, 80:237–251, 1973.

[Kim, 1993] Jaegwan Kim. Supervenience and mind. *Selected Philosophical Essays*, 1993.

[Kittur *et al.*, 2008] Aniket Kittur, H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proc. CHI*, 2008.

[Lappin and Leass, 1994] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Comput. Linguist.*, 20:535–561, 1994.

[Levin and Hovav, 1994] B. Levin and M. Rappaport Hovav. A preliminary analysis of causative verbs in english. *Lingua*, 92:35–77, 1994.

[Ling and Weld, 2010] Xiao Ling and Daniel S. Weld. Temporal information extraction. In *AAAI*, 2010.

[Liu and Singh, 2004] H. Liu and P. Singh. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22, 2004.

[Marneffe *et al.*, 2006] M. Marneffe, B. MacCartney, and C.D Manning. Generating typed dependency parses from phrase structure parses. In *Proc. LREC*, 2006.

[Miller, 1995] George A. Miller. Wordnet: A lexical database for english. *CACM*, 38:39–41, 1995.

[Schank, 1972] Roger Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology 3*, 4:552–631, 1972.

[Shahaf and Guestrin, 2010] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *Proc. KDD*, 2010.

[Suchanek *et al.*, 2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proc. WWW*, 2007.

[Wolff *et al.*, 2002] Phillip Wolff, Grace Song, and David Driscoll. Models of causation and causal verbs. In *Proc. ACL*, 2002.