Event Causal Relationship Retrieval

Yasunobu Sumikawa
Takushoku University
Tokyo, Japan
ysumikaw@cs.takushoku-u.ac.jp

ABSTRACT

Analyzing history has numerous benefits, including understanding what the people in the past did for events and what results they obtained and using historical knowledge to the present. Several past studies have analyzed historical events based on the assumption that each event is described in texts. Most of them analyze how similar the words and their categories used in the descriptions are instead of taking care of event-causal relationships.

In this study, we propose an algorithm named the Event Causality relationship similarity Measurement (ECM) to measure the similarity between event-causal relationships. The ECM solves a maximum weight matching problem on a bipartite graph, where the weights are the similarities between the event-causal relationships. We evaluated ECM with previous related works and confirmed that the ECM is the best.

CCS CONCEPTS

• **Information systems → Similarity measures**.

KEYWORDS

Event causal relationship, bipartite graph, a maximum weight matching problem, event retrieval

ACM Reference Format:
Yasunobu Sumikawa. 2021. Event Causal Relationship Retrieval. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI-IAT '21), December 14–17, 2021, ESSENDON, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3486622.3493936

1 INTRODUCTION

The importance of having a good knowledge of history and the ability to apply that knowledge in the present society has been gaining attention in recent years [1][4]. In fact, the importance of history is widely recognized, as many countries have classes for learning history from elementary school, and history curricula and learning support research emphasize the development of the ability to use knowledge in the present society rather than merely memorizing history [9]. As the saying goes, "History does not repeat itself, but rhythms repeat themselves," events occurring in the past and present often contain similarities as well as differences. In other words, to utilize knowledge from the past in the present, it is important to identify the similarity of not only the words used in the event descriptions, but also relationships such as cause-and-effect relationships [7].

Although there have been studies on retrieving events, most of them analyze how similar the words [9] and their categories [19] are used in the descriptions instead of taking care of event-causal relationships. In addition, several studies on history teaching and learning research fields have proposed methods that guide learners to measure the similarity between causal relationships [8]. However, the methods use fixed causal relationships defined by researchers; thus, it is difficult to use other relationships.

In this study, we propose an algorithm, named Event Causality relationship similarity Measurement (ECM), to measure the similarity of event-causal relationships. This study uses two assumptions: 1) each event-causal relationship includes at least two events that we call sub-events and 2) the sub-events are arranged in chronological order regarding when they occur. The proposed algorithm creates a bipartite graph, whose nodes and weighted edges represent sub-events and their similarities. It then measures the similarity between the two event-causal relationships by solving a maximum weight matching problem on the graph. We solve the matching problem by adding a constraint with *no intersection points on the edges that are the solutions to the problem.* This constraint allows us to evaluate the similarity between event-causal relationships.

Fig. 1 shows a motivation example. In this figure, there are event-causal relationships that represent three causality relationships, and there exists an event "Evacuation" at their end. Looking at the cause of the events, (a) and (b) are both "Landslides", but (c) is a "Typhoon." Because we search for similar causality relationships in this

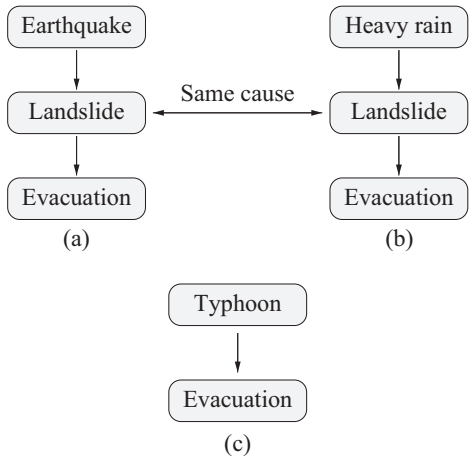

Figure 1: Example of causality relationship similarity measurement

WI-IAT '21, December 14–17, 2021, ESSENDON, VIC, Australia

ACM ISBN 978-1-4503-9115-3/21/12…$15.00
https://doi.org/10.1145/3486622.3493936

study, we consider that the two causality relationships represented by Landslide and Evacuation are more similar to each other than the causal relationships represented by Typhoon and Evacuation. Therefore, if the causal relationship of (a) is input, our algorithm outputs (b) as a result.

We evaluated ECM using a Wikipedia-based dataset named W2E, which included 322 topics and 1,041 events. We confirmed that ECM is the best algorithm in terms of MAP, precision, recall, and F_1-score compared with baselines.

The remainder of this paper is organized as follows. We present the definitions in Section 2 and the related work in Section 3. In Section 4, we describe the proposed algorithm. Section 5 presents the results of the experimental evaluation. The last section concludes the paper and describes future work.

2 DEFINITIONS

2.1 Sub-event, Event, and Causal relationship

We define a *sub-event* as a description of a single event. We also define an *event* as the event that oversees multiple *sub-events*. For example, if we consider this definition using World War II, the event is World War II and the sub-events include the attack on Pearl Harbor and the Potsdam Declaration. In this study, we define a *causal relationship* as a collection of two or more sub-events. Within this collection, we assume that the entire sequence can be defined in terms of the time series when sub-events occur.

2.2 Bipartite Graph

Let $G = (A, B, E)$ be the bipartite graph where A and B are two node sets and E is an edge set. Each node of A and B represents a sub-event. Weight of an edge (a_i, b_i) where $a_i \in A, b_i \in B$ represents the similarity between sub-events of the nodes a_i and b_i.

2.3 Ordered Bipartite Graph

An ordered bipartite graph G' defines the total order in two node sets A and B. That is, it defines the total order for all the subscripts i, j of $a_i, a_j \in A$.

2.4 Maximum Weight Matching

For a given graph G, matching is an edge set $M \subset E$ that does not share any nodes. Maximum weight matching is the problem of finding the highest sum of weights $M' \in M$ of the edges selected as M.

3 RELATED WORK

3.1 Event Classification and Detection

The most widespread research in analyzing event texts is classification. Kosmerlj *et al.* proposed event categories that were originally defined by Wikipedia editors, and then investigated automatic classification using Term frequency-inverse document frequency (TF-IDF) model created from news articles [10]. This study uses whole news article texts; however, several events are referred to in texts with several words. Sumikawa and Jatowt proposed a feature selection method for classifing short documents of past events [18]. Sumikawa and Ikejiri extended this study from single-label classification to multi-label classification [17]. These studies propose classification frameworks rather than search algorithms.

Several studies have proposed event detection methods based on texts. The method proposed by Manaskasemsak *et al.* detected emerging events from Twitter [13]. This method first identifies events by grouping similar tweets on a tweet graph whose nodes and edges represent tweets and their similarities. If the events are classified as trend lines, the method regards them as emerging events. The purpose of this study was to identify events instead of measuring the similarity between several events.

3.2 Causal Relationship Extraction

Causal relationships extraction is one of the most widely studied research topics. The problems that have been addressed in previous studies can be broadly divided into two categories: learning causal effects and learning causal relations [5]. Focusing on research using event texts, studies have been conducted to learn causal relationships from past events to predict possible future events. Radinsky and Davidovich proposed an algorithm to predict possible future events after mining causal relationship [15]. This algorithm first extracts causal relationships from the descriptions of past events. It then calculates the probability that an event will occur in the near future. As a causal relationship includes several events, causal relationship extraction is a task of topic detection and tracking. Looking at topic detection and tracking studies, there are methods to detect hot events from online news streams and track hot events [14], detect global hot events and local hot events using local community detection mechanisms [20], and reporting real-life events to users in the human-readable form [12].

These studies as well as our study focus on the causal relationship between events; however, our algorithm assumes that the causal relationship between events is already given and uses causal relationships to measure the similarity between events.

4 ECM ALGORITHM

In this section, we first describe the theory of ECM solving our maximum weight matching problem. We then describe dynamic programming that solves this problem.

4.1 Theory

In this study, we add a constraint with *no intersection points on the edges that are the solutions to the problem* to the general maximum weight matching problem. We assume that the graph is defined in a flat space. Fig. 2 shows the solutions obtained by solving the general maximum weight matching problem and the extended problem of this study. In the figure, the blue and red edges indicate the solutions of the general and extended maximum weight matching problem, respectively. Green edges represent the solutions shared by the two problems. Looking at the a_1, the node has two connected edges. The weight of the edge that is also connected to b_4 is higher than that of the other edge; thus, the blue edge is chosen as the solution in the general maximum weight matching. However, because it has intersections with the edges (a_2, b_2) and (a_3, b_3), it is not chosen as the solution in this study, and the other edge (a_1, b_1) is chosen as the solution.

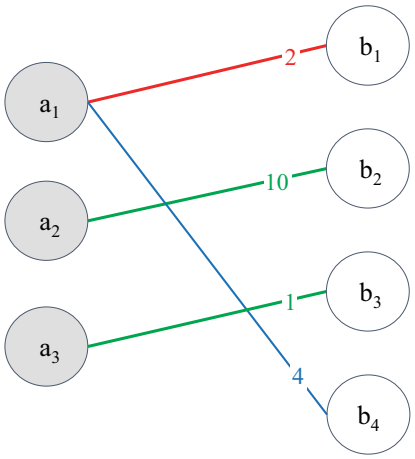

Figure 2: Example of solution difference between the general maximum weight matching problem and proposed problem in this study.

4.1.1 Ordered Bipartite Graph Construction. We assume that all nodes N are given before applying ECM. In addition, the given nodes are already divided into sub-sets respecting causal relationships and indexed respecting their chronological order about when their corresponding events occurred. For example, if an event a_1 causes another event a_2, then two nodes a_1, a_2 are defined and grouped in the same node set A.

We construct the ordered bipartite graph $G = (A, B)$ with the given node sets. To define edges, the algorithm uses the similarity between sub-events corresponding to the nodes taken from each set of A and B as weights of their edges.

4.1.2 Solving Proposed Maximum Weight Matching Problem on Ordered Bipartite Graph. The intuitive idea is that ECM measures similarity in the order when events have occurred from the past towards the present because the bipartite graph of this study defined in Section 4.1.1 already represents the causal relationship.

Let $e = (a_i, b_j)$ and $e' = (a_k, b_l)$ be the edges that are included in the solution ($e \neq e'$). In this case, we define the constraint *no intersection points on the edges that are the solutions of the problem* as follows.

- If $k < i$, then $l < j$.
- If $l < j$, then $k < i$.

The ECM selects edges that are candidates for the solution one by one in order from the one with the smallest index and calculates the sum of the weights of all the edges selected as the solution. It then chooses the edge set that maximizes the sum.

4.2 Implementation

In the remainder of this section, we describe the implementation of ECM as a dynamic programming solver. First, we explain how to fill a basic table in dynamic programming, and then present a detailed algorithm.

Figs. 3 and 4 show a graph and a table of an example of how to solve the constrained maximum weight matching problem on

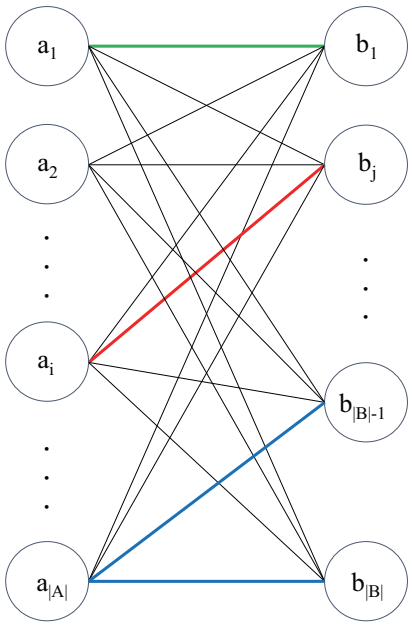

Figure 3: Example solutions of the proposed maximum weight matching problem

	1	2	3	⋯	j	⋯	|B|
1	70	66	99	57	56	76	94
2	2	18	73	10	82	69	3
3	27	26	13	96	79	89	22
⋯	58	85	54	38	46	67	30
i	8	55	14	78			
⋯							
|A|							

Figure 4: Dynamic programming solver

bipartite graphs by dynamic programming. The two figures color the edges/cells green, red, and blue if they are already analyzed, are currently being analyzed, and will be analyzed in the future, respectively. As the implementation is dynamic programming, the algorithm uses a table DP to record the sum of the weights of the selected edges. In addition, ECM records all the weights of the edges in a table W. ECM analyzes the nodes in A and B of a bipartite graph $G = (A, B)$ in ascending index order. If the edge currently being analyzed is selected as the solution, the sum of the weights is memorized in the table DP in addition to the set of edges previously selected as the solution. In other words, as shown by the arrows in Fig. 4, the weights of the selected edges from the green area are transferred to the red cells, and the edge with the highest value is finally selected as the solution. During the memorization on the table DP, once a cell is selected as one of the solutions, the

	b_1	b_2	b_3	b_4
a_1	2	0	0	4
a_2	0	10	0	0
a_3	0	0	1	0

(a) W

0	0	0	0	0
0	2	2	2	6
0	2	12	12	12
0	2	12	13	13

(b) DP

Figure 5: Solutions of Fig. 3 solved by dynamic programming

algorithm excludes all the cells not to select the same nodes more than twice. In addition, to satisfy the constraint that *edges do not have intersections*, the algorithm excludes cells with values smaller than one of the selected cells from the target such that they cannot be selected as solutions.

Fig. 5 shows the two tables W and DP, which record all the weights of the edges and the sum of the weights of the selected edges, respectively, that are created in this algorithm for Fig. 2. If there is no edge on the ordered bipartite graph, 0 is recorded in W. ECM records a value to the table DP by adding the highest of the three values from the left, upper left, and upper right of the table DP and the value of W corresponding to the current cell. This process is performed until all the cells have recorded their values. As a result, the sum of the weights of the edge set selected by this method can be recorded in the lower right corner of the table DP, and the actual edge set is obtained by scanning its component edges in the reverse direction.

In the following, we present the formal definitions of ECM. Let W be a matrix whose elements are the weights of each edge of the bipartite graph $G = (A, B)$. That is, using the function *weight* that returns the weight of the edge given as the argument, we define each element of W as $W_{i,j} = weight(e(a_i, b_j))$ for any $e_{a_i} \in A, e_{b_j} \in B$.

We then consider the process of recording values to table DP. Let TW_{ij} be a total weight by adding the weight of $e(a_i, b_j)$ and the solution edges whose indexes are from (0, 0) to $(i-1, j-1)$. From the definition of Section 4.1.2, TW_{ij} is larger than the total weights of the other edges connected to a_i or b_j selected as solutions. This is because the indexes of the nodes connected to the edges previously chosen as solutions are less than i and j, respectively. Thus, if the edge $e(a_i, b_j)$ is selected as the solution, the result of adding $W_{i,j}$ to $DP_{i-1,j-1}$ should be larger than $DP_{i,j-1}$ or $DP_{i-1,j}$; otherwise, the value is larger than the case where $e(a_i, b_j)$ is not the solution. The formal equation of this idea is as follows:

$$DP_{i,j} = max(DP_{i-1,j-1} + W_{i,j}, DP_{i-1,j}, DP_{i,j-1}) + W_{i,j} \quad (1)$$

Algorithm 1 ECM implemented as dynamic programming

Input: A weighted matrix W, tables DP and $rmax$
Output: A set of edge $SubE$

```
1:  Function ECM(W, DP, rmax)
2:    // Table calculation
3:    for i = 1 to ColumnSize(W)
4:      for j = 1 to RowSize(W)
5:        prev_max ← Val(rmax_{i-1,j-1})
6:        DP_{i,j} ← prev_max + W_{i,j}
7:        rmax_{i,j} ← tuple_val_max(rmax_{i-1,j}, rmax_{i,j-1})
8:        if DP_{i,j} > val(rmax_{i,j})
9:          rmax_{i,j} ← (DP_{i,j}, (i, j))
10:       end if
11:     end for
12:   end for
13:   // Answer determination
14:   i, j ← ColumnSize(W), RowSize(W)
15:   SubE ← ∅
16:   while i ≠ −1 and j ≠ −1
17:     SubE.append(e_{i,j})
18:     i, j ← Index(rmax_{i-1,j-1})
19:   end while
20:   return SubE
```

Algorithm1 shows the ECM algorithm[1]. ECM assumes that table DP is initialized with all the components of the first row and first column as 0. ECM also uses table $rmax$ to record the edges as solutions while filling in table DP. $rmax$ comprises a pair of a weight *value* of the edge to be solved and an index *index* of the connected nodes.

The 2nd~12th lines record the calculation results of Eq. 1 in each component of the table DP. *ColumnSize* (3rd line) is a function that returns the number of rows in the table of arguments. The function *RowSize* (4th line) returns the number of columns in the table of arguments. Looking at the 5th line, the *Val* function returns a value from a pair of values and subscripts given as arguments. The *tuple_val_max* on line 7 is a function that returns the largest value among the pairs of values and subscripts given as arguments. The function *Index* on line 18 returns indexes from a pair of the value and index given as arguments. During the filling DP, the algorithm records the subscripts of the selected edges as the solution in $rmax$. Because the indexes of the edges to be selected as solutions are arranged in ascending order, we can obtain the edge set required by this method by tracing the subscripts of $rmax$ in reverse order, as shown in lines 13–19.

Computational complexity. Because the DP matrix and $rmax$ can be calculated simultaneously, each $DP_{i,j}$ and $rmax_{i,j}$ can be obtained in constant time. In other words, the time calculation is $O(|A||B|)$. On the other hand, the spatial computation is $O(|A||B|)$ because the table size is determined by the number of nodes.

Theory. ECM obtains the optimum solution for $S, T \in \mathbb{N}$.

Proof. We prove that the algorithm obtains the optimal result by mathematical induction on the number of vertices of the bipartite graph $G = (A, B)$.

[1]The source code is available on our project repository: https://github.com/sumilab/programs/tree/master/ecm

Table 1: Example of test dataset. The first column represents topic categories of the second and forth columns. The second column is the topic ID of the third column's topic in W2E dataset. The forth and fifth columns are results of ECM if the third column's topic. Due to space limitations, some sentences and events are omitted.

Cat.	TOPIC ID	Event texts	TOPIC ID	Event texts
PE	TOPIC-896	2016-07-14 Elizabeth Truss is named Secretary of State for Justice and first ever female Lord Chancellor of the United Kingdom as former chancellor Michael Gove is ousted from the cabinet. 2016-07-13 The new Prime Minister of the United Kingdom Theresa May begins forming her ministry following the end of the Second Cameron ministry.	TOPIC-1780	2016-06-29 The process to elect a new leader of the Conservative Party to replace outgoing Prime Minister David Cameron begins in the United Kingdom. 2016-07-11 Prime Minister David Cameron announces he will step down on Wednesday, July 13. 2016-06-30 Former Mayor of London Boris Johnson rules himself out of running in the Tory leadership contest, a move believed to be influenced by Michael Gove's announcement earlier in the day to run for the leadership. 2016-07-05 Home Secretary Theresa May gets 165 votes after the first ballot of Conservative members of parliament to select a new Leader and the next Prime Minister.
S	TOPIC-1534	2016-09-10 In tennis, German Angelique Kerber defeats Czech Karolína Plíšková in three sets to win the 2016 US Open women's singles title. 2016-09-11 In tennis, Swiss Stan Wawrinka defeats Serbian Novak Djokovic in four sets to claim the 2016 US Open men's	TOPIC-1996	2016-01-31 In tennis, defending champion Novak Djokovic of Serbia defeats second seed Andy Murray of the United Kingdom in the men's singles in straight sets. It is Djokovic's third straight Grand Slam title. 2016-01-30 In tennis, Angelique Kerber of Germany tops (2–1) defending champion Serena Williams of the United States, 6-4, 3–6, 6–4, to win the Women's Singles. This is Kerber's first Grand Slam title.

Base case: When $|A| = |B| = 1$, the DP is $DP_{1,1}$. This indicates that the solution is to use only the edge (1, 1). It is trivial to use the edge as the solution. Therefore, ECM obtains the optimal result for $|A| = |B| = 1$.

Induction step: Let $S, T \in \mathbb{N}$ be given and suppose ECM obtains the optimal result for $S = |A|, T = |B|$. Consider the case of $|A| = S + 1$. We consider how to find the optimal solution when any edge connected to the node A_{S+1} is selected as the bottom side. At this time, the optimal solution must already be found for all the edges above the selected edge. From the condition, the optimal solution has already been found for all edges except the edge connected to the node A_{S+1}. Moreover, the other edges connected to node A_{S+1} are not the upper edges of the selected edge. Therefore, because the optimal solution is found for all the edges above the selected edge, the optimal solution may be found even when $|A| = S + 1$.

The optimum solution can be obtained for $|B| = T + 1$ in the same way.

5 EXPERIMENTAL EVALUATIONS

5.1 Experimental setting

5.1.1 Dataset. We used the W2E dataset [6] that contains events listed as a topic in this evaluation. The W2E dataset contains 3,083 topics consisting of 5,160 events recorded in Wikipedia's current event portal as of 2016. The events in the W2E dataset were collected from 10 categories of events, including the US presidential election, UK's European Union membership referendum, Middle East wars, Summer Olympics, and disasters in North America. These events were manually aggregated with multiple related events as topics. In addition, each topic has one of the following 10 categories: Sport (**S**), Armed conflicts and attacks (**AA**), Business and economy (**BE**), Science and technology (**ST**), Arts and culture (**AC**), Law and crime (**LC**), Politics and elections (**PE**), International relations (**IR**), Disasters and accidents (**DA**), and Health and medicine (**HM**). Tab. 1 shows examples of W2E dataset topic categories, topics, and their events.

As some W2E topics contain only one event, we used topics for more than one event to construct the bipartite graph in this study. Furthermore, we prepared pairs of similar topics before performing

Table 2: Statistics of test dataset.

Num. of events	1,041
Ave. num. of tokens	31.79
Num. of topics	322
Ave. num. of events per topic	3.23

Table 3: Statistics of test dataset by category.

	S	**AA**	**BE**	**ST**	**AC**
Ave. Num. of events	2.92	3.26	3.85	2.0	2.66
Num. of topics	13	73	7	2	3
Ave. Num. of tokens	29.44	27.15	34.59	23.25	34.37
	LC	**PE**	**IR**	**DA**	**HM**
Ave. Num. of events	3.06	3.14	3.43	3.11	4.66
Num. of topics	32	84	57	45	6
Ave. Num. of tokens	34.53	31.87	38.20	29.54	28.25

the experimental evaluation[2]. Two volunteers were asked if the pairs were similar or not. If both of them agreed that the pairs were similar to each other, we maintained them; otherwise, we discarded them. Tab. 2 shows the statistics of the subset of the W2E dataset used in this evaluation. In addition, Tab. 3 shows the statistics by topic category.

5.1.2 Baselines. We compare ECM with the following two methods.

- BM25: Okapi BM25 [16] is a widely used ranking function that ranks a set of documents based on the query terms appearing in each document.
- TF-IDF + dynamic time warping (DTW): DTW measures the similarity between time series data by finding the distance between each point of the two-time series in a brute force fashion and then finding the path where the two-time series are shortest after finding all of them.

TF-IDF indicates the importance of a word to a document in the dataset. This score is a multiplication of the term frequency and inverse document frequency. Term frequency refers to how frequently each term (word) occurs in each document whereas the inverse document frequency represents how rarely each term occurs in all documents. The formal definition is as follows.

$$TFIDF(w, d, \mathcal{D}) = tf_{w,d} * \frac{\mid \mathcal{D} \mid}{\mid \{d' \in \mathcal{D} \mid w \in d'\} \mid} \tag{2}$$

where $tf_{w,d}$ is the number of times a word w occurs in a document d, and $\mid \bullet \mid$ is the size of $\bullet$. The second term of this equation gives the number of all labeled data divided with labeled data including w.

5.1.3 Evaluation Criteria. In this paper, we evaluate the aforementioned algorithms and ECM by mean average precision (MAP), precision (P), recall (R), and F1-score (F_1). The MAP is calculated by considering the mean *AverageP*, which averages precision at K (ap@K) indicating precision among the top K documents. The

[2]The dataset is available in http://www.ysumi.sakura.ne.jp/sim_event_causalities.csv

Table 4: Similarity scores for feature creation. "-" indicates that no results were obtained correctly. The bold-faced numbers indicate the best for a particular term.

	TF-IDF	LSA	LDA	Doc2Vec
CosSim	**59.8**%	48.8%	26.4%	14.3%
Euclid	17.7%	11.2%	20.2%	15.0%
JS	14.0%	-	17.7%	-

formal equation of average precision is defined as follows:

$$Precision(k) = \frac{1}{k} \sum_{i}^{k} r_i \tag{3}$$

$$AverageP = \frac{1}{\mid \mathcal{D}^t \mid} \sum_{k<N} r_k Precision(k) \tag{4}$$

where r_i represents whether the prediction is correct or not using 1 (correct) or 0 (wrong), $\mathcal{D}^t$ is a set of test data and N is the last rank where a classifier assigns a correct label to the test data.

To calculate P, R, and F_1 scores, we use only k = 1, which represents the highest score of the results of applying each search algorithm.

5.2 Results

5.2.1 Bipartite graph creation. We first analyze feature vector creation and the measurement of similarity between the two vectors that are necessary to build a bipartite graph.

> **Q.** Which feature vector construction method and similarity evaluation method should be used to construct a bipartite graph to apply ECM?
> **A.** The combination of TF-IDF and cosine similarity is the best.

To create feature vectors, we used TF-IDF, latent semantic analysis (LSA) [3], latent Dirichlet allocation (LDA) [2], and Doc2Vec [11], which capture latent semantic text structures. LSA performs a matrix decomposition on the term-document matrix whereas LDA is a probabilistic model assuming a Dirichlet prior over the latent topics. Doc2Vec is a neural network-based algorithm.

Pairwise similarities of events are then computed using the cosine similarity (*CosSim*), Euclidean distance (*Euclid*), and Jensen-Shannon divergence (*JS*).

Tab. 4 shows the evaluation result of P (precision) for bipartite graph construction. These scores are the results of ECM indicating that each score represents the difference in how to build a bipartite graph whereas the topic search algorithm by dynamic programming is the same. From this result, we can observe that using TF-IDF and cosine similarity is the best for the graph construction. In the following, we use only the two methods for bipartite graph construction.

5.2.2 Retrieval results. In the remainder of this section, we analyze the retrieval results of the three algorithms.

Table 5: Evaluation results. The bold-faced numbers indicate the best for a particular term given the metric.

	MAP	P	R	F_1
BM25	33.1%	25.8%	25.8%	25.8%
DTW	54.4%	49.4%	49.4%	49.4%
ECM	**62.5%**	**59.8%**	**59.8%**	**59.8%**

Table 6: Correct retrieval results by ECM.

	S	**AA**	**BE**	**ST**
Num. of correct results	8	46	3	1
Ratio	61.54%	63.01%	42.86%	50.00%
	AC	**LC**	**PE**	**IR**
Num. of correct results	0	7	56	35
Ratio	0.00%	21.88%	66.67%	61.40%
	DA	**HM**		
Num. of correct results	26	6		
Ratio	57.78%	100.00%		

Q. Which is the best search algorithm among BM25, DTW, and ECM?
A. ECM is the best.

Tab. 5 shows the results for each algorithm. The results showed that the proposed algorithm obtained the best result. It also shows that the order of sub-events is important to consider when evaluating the similarity between events when considering causal relationships because all scores of the BM25 were the lowest among the three algorithms.

5.3 Error Analysis

In the following section, we discuss the results of analyzing only the proposed algorithm. In addition, we focus on the case of k = 1, which represents the result that the ECM is the most similar to the query.

Q. Which category of topics ECM searched most correctly?
A. All **HM** topics obtained the correct results. In addition, over 60% of the topics in the **PE**, **AA**, **S**, and **IR** categories yielded correct results.

At the beginning, we analyze the distribution of topic categories for which ECM has achieved correct results. This counts the number of matches between the categories of the topic given as the input and output by the ECM as k=1.

Tab. 6 shows the results. First, we can observe that ECM obtained the correct results for all topics in **HM**. **PE**, **AA**, **S**, and **IR** also yielded good results because more than 60% of the topics in their categories obtained correct results. However, many topics in **AC** and **LC** have missed obtaining correct results.

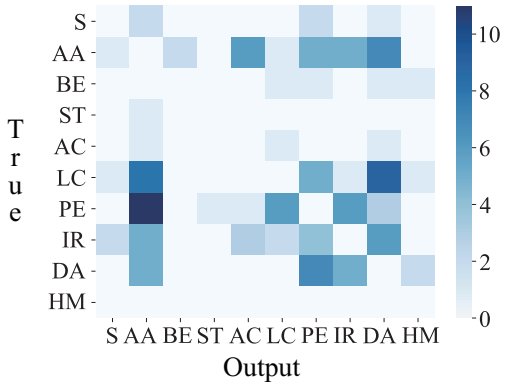

Figure 6: ECM wrong retrieving results

Q. Which category of topics was wrongly predicted for each category topic?
A1. PE topics were often misjudged as **AA**.
A2. LC topics were often misjudged as **AA** or **DA**.

Next, we analyze the results that the ECM wrongly determined to be the most similar. This counts which category the expected result topic category is wrong with.

Fig. 6 shows the results. In this figure, each row represents the expected result's topic category whereas each column indicates the topic category that resulted as the most similar.

As shown in the results, we can see that topics of **PE** were often misjudged as topics of **AA**. One possible reason for this misjudgment is that the number of topics in **AA** is the second largest number of topics. Compared with the number of topics between **AA** and others excluding **PE**, **AA** has approximately 2~37 times more topics than other categories. In addition, **LC** is often misjudged as either **AA** or **DA**. We discuss this point in the following analysis.

Q. Why were the results of **AC** and **LC** weak?
A1. LC topics tend to have relatively higher values of mutual information than topics in other categories.
A2. In **AC** and **LC** categories, the Jaccard coefficient scores between topics in each category are lower than the average.

Next, we explore why ECM obtained the wrong results for many topics in **AC** and **LC**. We first analyze the similarity of the topics using the Jaccard coefficient and mutual information (MI). The Jaccard coefficient counts the number of words shared between the two texts. MI measures the mutual dependence between the two texts. Their formal definitions are as follows:

$$Jaccard(A,B) = \frac{|\ T_A \cap T_B\ |}{|\ T_A \cup T_B\ |} \quad (5)$$

$$MI(A,B) = \sum_{a\in A}\sum_{b\in B} p(a,b)\log\left(\frac{p(a,b)}{p(a)p(b)}\right) \quad (6)$$

where: $|\ \cdot\ |$ is the size of the set and T_A and T_B are the tokens included in category A and B, respectively. The higher the score of the measurements, the more correlated they are.

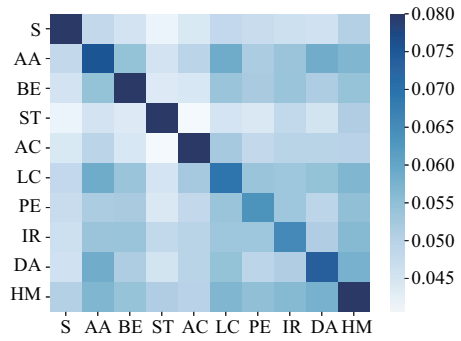

Figure 7: Topic similarity by Jaccard coefficient between categories

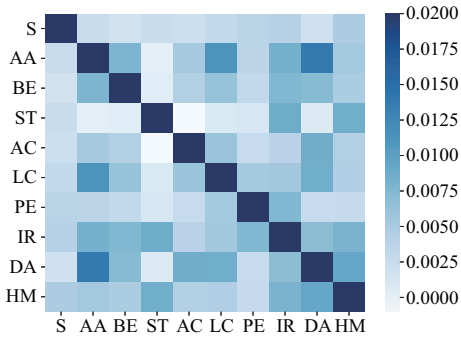

Figure 8: Correlation measure using Mutual Information between topics of different categories

Table 7: Topic similarity in the same category.

	S	**AA**	**BE**	**ST**	**AC**	
Jaccard	0.11	0.11	0.09	0.19	0.08	
MI	0.10	0.10	0.11	0.27	0.05	
	LC	**PE**	**IR**	**DA**	**HM**	Total
Jaccard	0.10	0.12	0.10	0.13	0.13	0.12
MI	0.09	0.13	0.10	0.13	0.10	0.12

Figs. 7 and 8 show Jaccard coefficients and MI scores between topics in different categories. As shown in Fig. 8, **LC** topics tend to have relatively higher values of mutual information between **AA** and **DA**. This is probably the reason why several **LC** topics were misjudged as topics in the two categories.

Next, Tab. 7 shows the Jaccard coefficient and MI scores of the topics in the same category. We can observe that both **AC** and **LC** are smaller than the overall average value in both scores; in particular, the MI scores of **AC** are the lowest among all categories. These results indicate that news descriptions in category **AC** rarely share the same words.

6 CONCLUSIONS

In this paper, we proposed an algorithm for measuring the similarity between events focusing on causal relationships. The algorithm solves a maximum weight matching problem with a novel constraint in which the edges that are solutions do not intersect on a bipartite graph whose nodes and edges represent events and their similarity, respectively. We implemented this algorithm and compared it to previous studies. We confirmed that the proposed algorithm obtained the best results.

There are three possible directions for future works. The first is to implement a search engine using this algorithm. The second is to generalize the problem by using causal relationships expressed in a tree structure. The third is to evaluate how this algorithm enhances the historical analogy of learners.

Acknowledgements. This work was supported in part by MEXT Grant-in-Aids (#19K20631).

REFERENCES

[1] Robert P. Abelson and Ariel Levi. 1985. Decision Making and Decision Theory, Handbook of Social Psychology. 231–309.
[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022.
[3] Scott Deerwester, Susan T. Dumais, George W. Furnas, Landauer Thomas K., and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.
[4] Thomas Gilovich. 1981. Seeing the Past in the Present: The Effect of Associations to Familiar Events on Judgments and Decisions. *Journal of Personality and Social Psychology* 40, 5 (1981), 797.
[5] Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. 2020. A Survey of Learning Causality with Data: Problems and Methods. *ACM Comput. Surv.* 53, 4, Article 75 (July 2020), 37 pages.
[6] Tuan-Anh Hoang, Khoi Duy Vo, and Wolfgang Nejdl. 2018. W2E: A Worldwide-Event Benchmark Dataset for Topic Detection and Tracking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1847–1850.
[7] Ryohei Ikejiri. 2011. Designing and Evaluating the Card Game which Fosters the Ability to Apply the Historical Causal Relation to the Modern Problems. *Japan Society for Educational Technology* 34, 4 (april 2011), 375–386. (in Japanese).
[8] Ryohei Ikejiri. 2011. Rekishi no ingakankei wo gendai ni ouyou suru tikara wo ikusei suru ka-do ge-mu kyouzai no dezain to hyouka. *Japan Society for Educational Technology* 34, 4 (2011), 375–386.
[9] Ryohei Ikejiri and Yasunobu Sumikawa. 2016. Developing World History Lessons to Foster Authentic Social Participation. *Journal of educational research on social studies* 84 (July 2016), 37–48.
[10] Aljaž Košmerlj, Evgenia Belyaeva, Gregor Leban, Marko Grobelnik, and Blaž Fortuna. 2015. Towards a Complete Event Type Taxonomy *(WWW '15 Companion)*. ACM, New York, NY, USA, 899–902.
[11] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents, Vol. 32. ICML'14, Bejing, China, 1188–1196.
[12] Zhen Lei, Ling-da Wu, Ying Zhang, and Yu-chi Liu. 2005. A System for Detecting and Tracking Internet News Event *(PCM'05)*. Springer-Verlag, Berlin, Heidelberg, 754–764.
[13] Bundit Manaskasemsak, Bodin Chinthanet, and Arnon Rungsawang. 2016. Graph Clustering-Based Emerging Event Detection from Twitter Data Stream *(ICNCC'16)*. ACM, New York, NY, USA, 37–41.
[14] Yajie Qi, Li Zhou, Huayou Si, Jian Wan, and Ting Jin. 2017. An Approach to News Event Detection and Tracking Based on Stream of Online News, Vol. 2. 193–196.
[15] Kira Radinsky and Sagie Davidovich. 2012. Learning to Predict from Textual Data. *J. Artif. Int. Res.* 45, 1 (Sept. 2012), 641–684.
[16] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST, 109–126.
[17] Yasunobu Sumikawa and Ryohei Ikejiri. 2021. Feature selection for classifying multi-labeled past events. *Int. J. Digit. Libr.* 22, 1 (2021), 63–83.
[18] Yasunobu Sumikawa and Adam Jatowt. 2018. Classifying Short Descriptions of Past Events *(ECIR '18)*. Springer International Publishing, 729–736.
[19] Yasunobu Sumikawa and Adam Jatowt. 2018. System for Category-driven Retrieval of Historical Events *(JCDL '18)*. ACM, New York, NY, USA, 413–414.
[20] Zhicong Tan, Peng Zhang, Jianlong Tan, and Li Guo. 2014. A Multi-layer Event Detection Algorithm for Detecting Global and Local Hot Events in Social Networks. *Procedia Computer Science* 29 (2014), 2080–2089. 2014 International Conference on Computational Science.