EventMiner: Mining Events from Annotated Documents

Dhruv Gupta[*†] Jannik Strötgen[*] Klaus Berberich[‡*]

[*]Max Planck Institute for Informatics [†]Saarbrücken Graduate School for Computer Science [‡]htw saar
Saarbrücken, Germany

dhgupta@mpi-inf.mpg.de, jstroetge@mpi-inf.mpg.de, kberberi@mpi-inf.mpg.de

ABSTRACT

Events are central in human history and thus also in Web queries, in particular if they relate to history or news. However, ambiguity issues arise as queries may refer to ambiguous events differing in time, geography, or participating entities. Thus, users would greatly benefit if search results were presented along different events. In this paper, we present EventMiner, an algorithm that mines events from top-k pseudo-relevant documents for a given query. It is a probabilistic framework that leverages semantic annotations in the form of temporal expressions, geographic locations, and named entities to analyze natural language text and determine important events. Using a large news corpus, we show that using semantic annotations, EventMiner detects important events and presents documents covering the identified events in the order of their importance.

Keywords

Information Retrieval; Text Mining; Semantic Annotations.

1. INTRODUCTION

Typically, events happen during particular time intervals at multiple locations. In previous works, it was shown that these two dimensions of events are also very frequent in web queries. In particular, 13.8% and 17.1% of web queries are explicitly and implicitly time-sensitive in nature, respectively [38]. Similarly, 12.7% of queries in *Yahoo!* query logs were reported to contain a geographic location [40]. Thus, we claim that events could be an important feature to navigate the Web. We show that events can be used as user intents to improve retrieval effectiveness. For this, we define that an event involves multiple named entities, at multiple locations, during multiple time intervals. An example of a textually realized event, which will be explained in detail in the next section, is presented in Figure 1. Due to marked accuracy and scalability of tools that provide semantic annotations in the form of temporal expressions, geographic locations, and named entities, the large-scale detection of events has become possible in the last years.

ICTIR '16, September 12-16, 2016, Newark, DE, USA

 ISBN 978-1-4503-4497-5/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2970398.2970411

> Cold war was a conflict involving Soviet Union⟨GEO:Soviet_Union⟩ and the US⟨GEO:USA⟩ under the presidency of Mikhail Gorbachev⟨WIKI:Mikhail_Gorbachev⟩ and Ronald Reagan⟨WIKI:Ronald_Reagan⟩ respectively, during late 1980s⟨01−01−1985,31−12−1989⟩.

Figure 1: Sample text with semantic annotations.

The understanding of time and space is essential to detect events. Temporal expressions can be highly uncertain (e.g., `late 1980s`) and ambiguous (e.g., `last spring`). Mentions of locations in text can also be highly uncertain and thus difficult to anchor on a map (e.g., Springfield). A document may talk about a single event in entirety or may shift its focus between different events. Considering these issues in addition to named entities increases the challenges manifold to mine and determine the importance of events from text. Prior approaches [3, 17, 24, 26] have tried to solve somewhat similar tasks albeit disregarding many of the issues outlined above when modeling an event.

In this paper, we describe EventMiner, a probabilistic framework which leverages text and multiple kinds of semantic annotations to mine important events. EventMiner is based on the family of *Dirichlet process mixture* models to cluster *semantically* similar text to mine events. The notion of *importance* in our algorithm is measured by statistical frequency. In order to analyze these annotations, we present simple mathematical models for complex notions of time and space that can aid in computing these similarities efficiently and effectively. Our model of time takes into account temporal uncertainty as well as temporal proximity in order to identify events that might occur very close to each other on a timeline (e.g., `1990` is close to `1991`). We additionally model geographic locations so that events happening at locations that neighbor each other are also captured (e.g., USA neighbors Canada). We additionally take into account similarity behind named entities (e.g., Ronald Reagan is related to Nancy Reagan) when identifying the important events. We evaluate our methods by comparing the summary of events computed for *history-oriented* queries [11] with their *Wikipedia*[1] page.

Applications. In many scenarios, e.g., in the context of *digital humanities*, there is a need for tools to analyze large text collections. Given keyword-only queries related to (historical) entities or events, EventMiner is able to mine these important events which can be further used to explore the retrieved documents.

[1]https://en.wikipedia.org/

Contributions. Our major contribution, EventMiner, is an algorithm that mines important events from semantically annotated text. In doing so, we present a novel time model that incorporates *proximity* between uncertain temporal expressions.

Outline. We first formally define the problem setting in Section 2. We describe computational models for annotations in Section 3 and EventMiner in Section 4. Our evaluation setup and experimental findings are given in Section 5. Related work is surveyed in Section 6. Conclusions drawn from the study are summarized in Section 7.

2. PRELIMINARIES

In this section, we describe and formally define the various semantic annotations of text that we use, how we preprocess, and index the data. We also formally put forward the hypothesis and the problems we solve in this article.

2.1 Semantic Annotations

Consider the piece of text with semantic annotations in Figure 1 as an illustrative example.

Temporal Expressions, as natural language in general, are highly ambiguous in nature. They can be categorized as *explicit*, *implicit*, *relative*, and *underspecified* [7, 12, 27]. An explicit expression in Figure 1 is `late 1980s`. As all temporal expressions, explicit expressions can be of different granularities, e.g., `May 5, 2001` or `1992`. Implicit expressions may not be immediately identifiable as they are characterized by words that carry a latent temporal meaning, e.g., `Christmas`. Words whose temporal meaning can only be resolved with respect to some other time point (e.g., the publication date) are known as relative (e.g., `yesterday`) or underspecified if the relation to the reference time has to be determined additionally (e.g., `April`). Tools that perform the complex task of extracting and normalizing such types of temporal expressions are HeidelTime [27] and SUTime [8], which we used in this work to resolve temporal expressions.

Geographic Locations and Other Named Entities. Detection and disambiguation of named entities is a non-trivial task. We use Aida [14] to identify, disambiguate and link named entities in text to an external ontology. Aida performs *named entity recognition and disambiguation* by leveraging statistical popularity of named entities and a contextual similarity algorithm to disambiguate them. Examples of named entities in Figure 1 are Mikhail Gorbachev and Ronald Reagan which have been disambiguated and linked to the *Wikipedia* identifier in the Yago ontology [29]. Geographic locations mentioned in text are called *toponyms* [25]. The process of resolving these toponyms to a specific location is known as *toponym resolution*. We use the geographical locations obtained as part of the detected and disambiguated named entities by Aida. Examples of disambiguated locations in Figure 1 are Soviet Union and US.

Document Collection. We used The New York Times Annotated corpus[2] which consists of around two million news articles published between 1987 and 2007. We utilized a news archive for evaluating our methods since it has been shown in prior work [11, 19] that they cover historic events very well. All the annotations along with text are preprocessed using the Hadoop map-reduce framework and subsequently indexed using the ElasticSearch[3] software.

[2] https://catalog.ldc.upenn.edu/LDC2008T19
[3] https://www.elastic.co/

2.2 Problem Statement

Consider a document collection D consisting of N documents d:

$$D = \{d_1, d_2, \ldots d_N\}$$

further each document $d \in D$ consists of sentences s:

$$d = \langle s_1, s_2, \ldots s_n \rangle$$

Each sentence s then contains a multiset of temporal expressions $s_{\mathcal{T}}$, geographic locations $s_{\mathcal{G}}$, named entities $s_{\mathcal{E}}$, and words $s_{\mathcal{W}}$ from a vocabulary $\mathcal{V}$:

$$s = \langle s_{\mathcal{T}}, s_{\mathcal{G}}, s_{\mathcal{E}}, s_{\mathcal{W}} \rangle.$$

The cardinalities of these multisets are given by $|s_{\mathcal{T}}|$, $|s_{\mathcal{G}}|$, $|s_{\mathcal{E}}|$, and $|s_{\mathcal{W}}|$. The aim is to design an algorithm:

$$\textsc{EventMiner}(S, Q, \Lambda),$$

where, S is a set of input sentences, Q is a keyword query, and Λ consists of a set of parameters$\Lambda \in \mathbb{R}^m$. The input set of sentences S is obtained from the pseudo-relevant set of documents R obtained via an information retrieval engine using the keyword query Q:

$$R = \textsc{Ir}(D, Q, K, \Theta),$$

where Θ is set of parameters and K specifies the number of documents to be returned by the retrieval method. The algorithm should output a totally ordered set of events:

$$\mathcal{C} = \langle c_1, c_2, \ldots c_k \rangle,$$

where c_i is an event. The ordering of events in $\mathcal{C}$ is done by using the scores obtained for each cluster given by the EventMiner algorithm. Using $\mathcal{C}$ we re-rank R to obtain a set of documents $\hat{R}$ so that the user sees at least one document from each event $c \in \mathcal{C}$.

An *event* that can be detected in text, is defined to involve multiple named entities $c_{\mathcal{E}}$, occurring during a time interval $[b, e] \in c_{\mathcal{T}}$ at locations $g \in c_{\mathcal{G}}$, and described by words $c_{\mathcal{W}}$. Thus each event $c \in \mathcal{C}$:

$$c = \langle c_{\mathcal{T}}, c_{\mathcal{G}}, c_{\mathcal{E}}, c_{\mathcal{W}} \rangle.$$

We hypothesize that using events as proxies for user intents we can increase the retrieval effectiveness of traditional information retrieval methods, which have largely relied on term statistics [11].

2.3 Assumptions

We make the following assumptions:

- Each semantic annotation occurs independent of each other. Hence, we can consider a sentence to contain a multiset of temporal expressions $s_{\mathcal{T}}$, geographic locations $s_{\mathcal{G}}$, and named entities $s_{\mathcal{E}}$.
- A multiset of geographic locations $s_{\mathcal{G}}$ or named entities $s_{\mathcal{E}}$ can be empty. However, they cannot be empty *simultaneously* in a sentence. A multiset of temporal expressions $s_{\mathcal{T}}$ in a sentence *cannot* be empty. In case no temporal expression occurs in a sentence we utilize the document publication date.
- Geographic annotations are a subset of named entity annotations, that is $s_{\mathcal{G}} \subseteq s_{\mathcal{E}}$.

3. COMPUTATIONAL MODELS

Here we describe the computational models used to represent and compute similarities between annotations for time, space, and named entities.

3.1 Time Model

We next explain two time models that we use in the EventMiner algorithm. Figure 2 illustrates these models.

Uncertainty-aware Time Model (Utm). Utm is a time model for uncertain temporal expressions [5], for instance `1990s`, where begin and end of the time interval $[b, e]$ cannot be clearly identified. Utm models this uncertainty by allowing for a *lower* and *upper* bound in the start (end) of the interval. Formally, a temporal expressions is modeled as a four-tuple:

$$t = \langle b_\ell, b_u, e_\ell, e_u \rangle,$$

where $b \leq e$, $b_\ell \leq b \leq b_u$ and $e_\ell \leq e \leq e_u$. The elements of t are obtained from a time domain $\mathbb{T}$. Thus, $[b, e] \in \mathbb{T} \times \mathbb{T}$. Hence, `1990s` is represented as $\langle 1990, 1999, 1990, 1999 \rangle$. The number of time intervals that can be generated from a temporal expression is denoted by $|t|$. The probability of generating a time interval $[b, e]$ from temporal expression t is estimated as [5]:

$$P([b, e]|t) = \frac{\mathbb{1}([b, e] \in t)}{|t|}.$$

Thus the likelihood of generating t' from t is:

$$P(t'|t) = \frac{1}{|t'|} \sum_{[b,e] \in t'} P([b, e]|t). \quad (1)$$

Following, Utm, we can for example, generate the time interval $[1995, 1998]$ from the temporal expression $\langle 1990, 1999, 1990, 1999 \rangle$. However, a time interval $[2000, 2001]$ will receive zero probability given the same temporal expression.

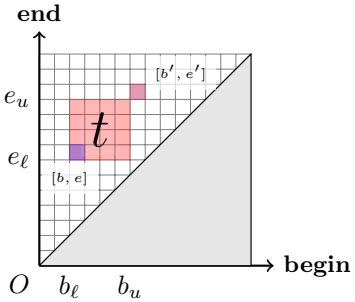

Figure 2: Graphical representation illustrating how a time interval $[b, e]$ is generated from t using Utm. It also represents graphically how the time interval $[b', e']$ obtains zero probability from Utm but a non-zero probability using Ptm.

Proximity-aware Time Model (Ptm). In Utm a time interval $[b, e] \notin t$ obtains zero probability. However, time intervals that are *temporally close* to time intervals in t should obtain non-zero probability [28]. This is required for computing similarity between events that have close but non-overlapping time intervals of occurrence (e.g., `1990` and `1991`). We compute the *proximity* by multivariate kernel density estimates. Concretely,

$$P([b, e]|t) = \frac{1}{|t|} \sum_{[b',e'] \in t} \mathcal{K}_A([b, e] - [b', e']),$$

where, $\mathcal{K}_A$ is a multivariate kernel estimator with bandwidth matrix A. The difference between time intervals is carried out element-wise, i.e., $[b - b', e - e'] = [b, e] - [b', e']$. The kernel density estimator $\mathcal{K}_A$ is defined as [13]:

$$\mathcal{K}_A(\bullet) = \frac{1}{|A|} \mathcal{K}\left(A^{-1} \bullet\right),$$

where, the bandwidth matrix A is:

$$A_{2\times 2} = k \cdot I_{2\times 2} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix},$$

where, $|A|$ represents the matrix determinant and A^{-1} its inverse. Proximity between t' and t can be estimated by:

$$P(t'|t) = \frac{1}{|t'|} \sum_{[b,e] \in t'} \frac{1}{|t|} \sum_{[b',e'] \in t} \mathcal{K}_A([b, e] - [b', e']). \quad (2)$$

For $\mathcal{K}_A(\bullet)$, we utilized the Epanechnikov kernel, since its support is $[-1, 1]$ and density at a point is computed using the values lying in the cube surrounding it [13]. It can be written as [13]:

$$\mathcal{K}_A(v) = \frac{3}{4}(1 - v^T v)\mathbb{1}(|\sqrt{v^T v}| \leq 1),$$

where $v = [b, e]$ is a time interval and the indicator function $\mathbb{1}(\bullet)$ evaluates to one iff the argument is true. The proximity between two temporal expressions can be varied by scaling k in the bandwidth matrix A.

The proximity model thus allows us to associate a non-zero probability with non-overlapping temporal expressions. Thus, the time interval [2000, 2001] can now be generated from the temporal expression $\langle 1990, 1999, 1990, 1999 \rangle$.

3.2 Space Model

Each geographic location g in our model is represented as a *minimum bounding rectangle* (MBR) with coordinates for its lowermost coordinate ℓ and uppermost coordinate u: $g = \langle \ell, u \rangle$.

Each coordinate lies in a two dimensional geodesic space, that is $\ell = (x_\ell, y_\ell)$ and $u = (x_u, y_u)$. For any two geographic locations described by their MBRs we can find similarity by computing the *area overlap* in the geodesic space. Similarly proximity can be found by their closeness in the geodesic space. Using the geodesic system to represent MBRs helps in avoiding distortion along the poles. Figure 3 shows how a MBR for `UK` is obtained from its set of coordinates.

(a) Plot of coordinates of UK. *(b) Plot of the MBRs of UK.*

Figure 3: Depiction of how MBRs are obtained from a set of geographic coordinates. For each dense region of coordinates we compute a MBR.

For an efficient indexing and querying of geographic locations, we utilize an R-Tree [36]. Each location's MBRs are indexed in an R-Tree. We can subsequently query the R-Tree for containment or proximity queries.

3.3 Entity Model

Each entity that is disambiguated by AIDA [14] is linked to the YAGO [29] ontology. Each YAGO entity is in turn linked to its Wikipedia entry. To compute the similarity between two entities e and e' we thus look at the *relatedness* in terms of Wikipedia links. This similarity is known as *Milne-Witten* entity-entity relatedness — $\text{MWSIM}(e, e')$ [34]. Formally, with W_e and $W_{e'}$ being the sets of articles that are linked to the Wikipedia articles corresponding to the entities e and e', respectively. Let W represent all articles in Wikipedia; the similarity is computed as [34]:

$$\text{MWSIM}(e, e') = \frac{log(max\,(|W_e|, |W_{e'}|)) - log\,(|W_e \cap W_{e'}|)}{log\,(|W|) - log\,(min\,(|W_e, W_{e'}|))}$$

4. EVENTMINER ALGORITHM

The EVENTMINER algorithm is a probabilistic model that takes into account the similarity between the semantic annotations in order to mine important events. It is based on the family of *Dirichlet process mixture* (DPM) models [4, 23, 37]. The rationale behind using DPM models is two-fold. First, they allow to *jointly* model the marginal distributions underlying the different dimensions for the events. Second, they allow to model an *infinite* number of clusters without knowing their number apriori. Each cluster identified by EVENTMINER is treated as an important event for the given keyword query. We next describe how the various semantic similarities are considered together, and how to perform inference over the probabilistic model.

Generating Sentences. Let $\mathcal{V}$ be the vocabulary associated with the document collection. Further, let the event clusters that are identified by EVENTMINER be described by a multinomial θ_c distributed over $\mathcal{V}$. We describe the probability of generating the sentence s_i given θ_c as [23, 37]:

$$P(s_i|\theta_c) = \prod_{v \in \mathcal{V}} \theta_c(v)^{tf(v,s_i)},$$

where the probability of obtaining the term v from cluster c is denoted by $\theta_c(v)$ and the term frequency of v in sentence s_i is denoted by $tf(v, s_i)$.

Given a term distribution over sentences, we now consider the similarity between the various semantic annotations.

Temporal Similarity. Let us consider an existing event cluster c, for a sentence containing a multiset of temporal expressions $s_\mathcal{T}$ to be similar in the temporal domain to c, we need to compute the temporal similarity. This is done by considering the temporal similarity of $t \in s_\mathcal{T}$ with all the temporal expressions $t' \in c_\mathcal{T}$ in the event cluster c as:

$$w_t(s, c) = \frac{1}{|c_\mathcal{T}|} \sum_{t' \in c_\mathcal{T}} \frac{1}{|s_\mathcal{T}|} \sum_{t \in s_\mathcal{T}} \mathbb{1}(t \in t'),$$

where the function $\mathbb{1}(t \in t')$ indicates likelihood of generating the temporal expression t from t' by Eq. (1) or (2).

Geographic Similarity. Similarly, given an event cluster c, to consider the similarity of a sentence along the spatial dimension we compute it as:

$$w_g(s, c) = \frac{1}{|c_\mathcal{G}|} \sum_{g' \in c_\mathcal{G}} \frac{1}{|s_\mathcal{G}|} \sum_{g \in s_\mathcal{G}} \mathbb{1}(g \in g'),$$

where the function $\mathbb{1}(g \in g')$ indicates the likelihood of generating the geographic location g from the geographic location g'.

Entity Similarity. On similar lines, we can compute the similarity between named entities in a sentence with an event cluster c as follows:

$$w_e(s, c) = \frac{1}{|c_\mathcal{E}|} \sum_{e' \in c_\mathcal{E}} \frac{1}{|s_\mathcal{E}|} \sum_{e \in s_\mathcal{E}} \mathbb{1}(e \sim e'),$$

where the function $\mathbb{1}(e \sim e')$ computes the relatedness between the entities e and e' using $\text{MWSIM}(\bullet)$. The motivation for computing the average entity-entity relatedness (as compared to maximum entity-entity relatedness) between sentence s and cluster c is to maintain the cluster coherence.

Joint Similarity. We combine all the semantic similarities in a weighted average. This can be written as:

$$w(s, c) = \frac{\rho_1 \cdot w_t(s, c) + \rho_2 \cdot w_g(s, c) + \rho_3 \cdot w_e(s, c)}{\rho_1 + \rho_2 + \rho_3}.$$

Chinese Restaurant Process. To incorporate the different semantic similarities in the Dirichlet process mixture model framework, we need to consider the *clustering* behavior of Dirichlet processes [31]. We briefly explain the framework next, following its description in [31, 33].

Consider a Dirichlet process (DP), $G_0 \sim DP(\alpha, H_0)$, with a prior (base) distribution H_0 [31, 37]. Further let the prior follow a Dirichlet distribution, $H_0 \sim Dir(\beta m)$; where, m is the normalized term frequency vector over $\mathcal{V}$ and β is a hyperparameter [18, 31, 37, 39]. Let, the sample multinomial distributions be drawn *iid* from G_0 be $\theta_1, \theta_2, \ldots, \theta_i$. For the predictive distribution θ_{i+1} it has been shown in [6, 31, 33] that:

$$\theta_{i+1}|\theta_1, \ldots, \theta_i, \alpha, H_0 \sim \sum_{\ell=1}^{i} \frac{1}{i + \alpha} \delta_{\theta_\ell} + \frac{\alpha}{i + \alpha} H_0,$$

where, δ_{θ_ℓ} is a Dirac delta at θ_ℓ. We can clearly see that if a sample is drawn more than once, it shall have a higher probability of being drawn again leading to a "positive reinforcement effect" or equivalently "rich gets richer effect" from the above equation [31, 33, 39]. This phenomenon is more often described in the literature as the *Chinese Restaurant Process*. The analogy is described as follows, consider unlimited tables to sit at a Chinese restaurant [31]. A new customer will sit at a table in the restaurant with probability that is dependent on the count of customers already occupying that table [31]. It can also be proven that the number of clusters in a Dirichlet process grows logarithmically [31].

Following this framework, we can incorporate the probability of assigning sentence s_i to cluster c, given all the other event cluster assignments as follows:

$$P(z_i = c|z_{-i}) = \begin{cases} \frac{w(s_i, c)}{\sum_{c'} w(s_i, c') + \alpha}, & \text{if } c \text{ is in cluster set} \\ \frac{\alpha}{\sum_{c'} w(s_i, c') + \alpha}, & \text{if } c \text{ is a new cluster} \end{cases}$$

where α is the concentration parameter.

Inference. To infer the cluster label z given the semantically annotated text, we can use the following inference:

$$P(z_i = c|z_{-c}, S) \propto P(z_i = c|z_{-i}) P(s_i|s_{-i} \in z_i = c),$$

where z_{-c} is used to denote all the cluster assignments except z_c, and $s_{-i} \in z_i = c$ denotes all the sentences in cluster c except s_{-i} [23, 37]. Since each type of expression is associated with a multiset, the order, in which semantic annotations are observed, can safely be ignored. Thus the first term, $P(z_i|z_{-i})$, can be computed by taking z_i as last in the order [23, 31].

The second term, $P(s_i|s_{-i} \in z_i = c)$, can be computed with the help of the Dirichlet process described earlier. It has been shown to be equal to [18, 23, 37]:

$$P(s_i|s_{-i} \in z_i = c) = \int p(s_i|\theta)p(\theta|s_{-i} \in z_i = c)d\theta$$
$$= \left(\frac{\Gamma(\sum_v tf(v, s_{-i} \in z_i = c) + \beta)}{\prod_v \Gamma(tf(v, s_{-i} \in z_i = c) + \beta m_v)}\right)$$
$$\left(\frac{\prod_v \Gamma(tf(v, s_i) + tf(v, s_{-i} \in z_i = c) + \beta m_v)}{\Gamma(\sum_v tf(v, s_i) + \sum_v tf(v, s_{-i} \in z_i = c) + \beta)}\right)$$

where the Gamma function is denoted by $\Gamma(\bullet)$, the term frequency of v in the set of sentences (excluding s_i) in cluster c is given by $tf(v, s_{-i} \in z_i = c)$, and the term frequency of v in sentence s_i is given by $tf(v, s_i)$.

The graphical model corresponding to our EventMiner algorithm, is illustrated in Figure 4. We utilized a modified Gibbs Sampler based on [37] for a single machine implementation. Similar to their implementation, our modified Gibbs Sampler also has time complexity of $\mathcal{O}(|S|^2)$.

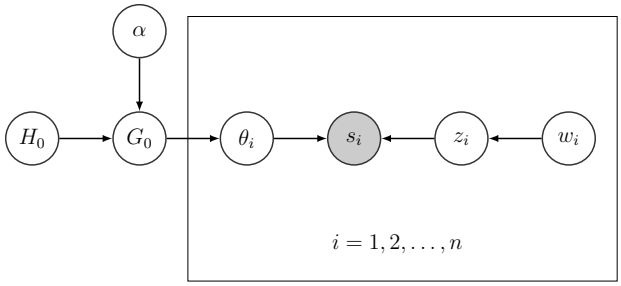

Figure 4: Graphical representation of the EventMiner *algorithm based on [37]. It shows a Dirichlet process G with concentration parameter α and its base measure H_0. Where, a sentence is denoted by s, its cluster label is denoted by z, and the joint similarity incorporating time, geographic location, and named entities is denoted by w. The duplicated random variables are represented in the box; with i denoting the multiplicity. Observed node is shaded in gray.*

Exploring Search Results with Event-Clusters. For each event-cluster c we can associate the documents from which the sentences were derived. This can thus be used to explore the search results in an event-centric manner.

5. EVALUATION

In this section we first describe the experimental setup and then discuss the results of the experiments.

5.1 Experimental Setup

We describe the experimental setup of our evaluation next.

History-Oriented Queries. In order to test the effectiveness of our method, we utilize history-oriented queries that have multiple important events associated with them. Specifically, we considered three types of query categories, namely: events [9], entities[4], and wars [20]. The keywords corresponding to these queries are shown in Table 1. This dataset of history-oriented queries and corresponding *Wikipedia* articles has been made publicly available as part of the research work carried out by Gupta and Berberich [11].

[4]http://usatoday30.usatoday.com/news/top25-influential.htm

Americas | american civil war | american revolution | mexican revolution
Europe | world war II | world war I | french revolution | punic wars | spanish civil war | russo-polish war
Africa | french algreian war | biafran nigerian civil war
Asia | vietnam war | korean war | iraq war | persian wars | chinese civil war | iran iraq war | russian civil war | french indochina war | russo-japanese car

(a) Wars.

Sports | commonwealth games | asian games | summer olympics | winter olympics | super bowl winners
Music | u2 album | nirvana album | beatles album | red hot chilli peppers album | michael jackson album
Movies | harry potter movie | oscar academy awards | lord of the rings movie
Politics | german federal elections | us presidential elections | australia federal elections

(b) Events.

Business | bill gates | sergey brin | larry page | howard schultz | sam walton
Science | stephen hawking | francis collins | craig venter
Politics | ronald reagan | mikhail gorbachev | george w. bush | deng xiaoping | nelson mandela | bill clinton | hillary clinton
Arts | j. k. rowling | oprah winfrey | russell simmons | bono
Religion | pope john paul II
Sports | lance armstrong | michael jordan
Other | ryan white | homer simpson | osama bin laden

(c) Entities.

Table 1: History-oriented queries [11].

Metrics. For each history-oriented query, our aim is to detect as many important events as possible. This task can be considered equivalent to producing a *summary* for a given topic. However, in our setting we disregard grammatical structure or temporal ordering. Rather our focus is specifically on *recall* and *precision* of information. We consider the frequent words in each event cluster $\mathcal{W}$ as a sentence and concatenate them from some top-ranked clusters to form a *system-generated summary*. In order to test this objectively, we utilized the Rouge-N measure [35] for evaluating the quality of summaries produced by our methods. We used the *Wikipedia* entry corresponding to the history-oriented query as the gold standard summary. Rouge-N then computes the overlap of n-grams between the gold standard summary and the summary produced by the system under test. We report the Rouge-N recall, precision and F_1.

Systems. We compare three systems. For them, we consider the EventMiner algorithm incorporating semantic similarity with increasing sophistication. As a naïve baseline, EventMiner computes similarities of various semantics by only considering *surface-level* equivalence. That is two semantic annotations are considered to be similar iff their tokens match. For example, 1995 and 1990s are *not* similar to each other. We call this system EMBaseline. For the second system, we compute the temporal similarity using uncertainty-aware time model Utm; the geographic similarity by computing area overlap in MBRs; and entity-entity similarity with MWSim. Therefore, in this system, 1995 and 1990s are similar; as are New York and USA; and

also Ronald Reagan and Nancy Reagan. We call this system EMSimilar. For the final system we compute the similarity between temporal expressions using proximity-aware time model Ptm; the geographic similarity by computing closeness between MBRs; and entity-entity similarity with MWSim. Therefore, this system considers 1999 and 2000 proximate; also Canada and USA are considered to be proximate. We call this system EMProximity.

5.2 Experiments

In this section, we describe the results obtained for the evaluation setup presented earlier.

Parameter Tuning. We performed the experiments by retrieving top-25 pseudo-relevant documents, i.e., $K = 25$ for every keyword query in the testbed. We choose this as a conservative estimate for the number of highly relevant documents given the keyword query. For retrieving these documents, we used the state-of-the-art *Okapi BM25* method with standard parameter settings $k_1 = 1.2$ and $b = 0.75$. These documents are subsequently split at sentence-level granularity.

The different EventMiner systems are next executed with these sentences as input. Weights for different similarities are: $\rho_1 = 0.50$, $\rho_2 = 0.25$, and $\rho_3 = 0.25$, giving more importance to temporal similarity as compared to the other two similarities. This was due to the observation that annotations for time were more accurate as compared to named entity annotations. Hence, temporal similarity was given a higher weight in computing joint similarity to obtain coherent clusters. Further, the concentration parameter was set to $\alpha = 0.1$ and the strength of prior for text similarity $\beta = 0.1$. The concentration parameter is directly proportional to the probability of creating a new event cluster. Both these values were set by observing their effect on three sample queries, namely: summer olympics, us presidential elections, and george w bush. We perform Gibbs Sampling for a total of 50 iterations, with early termination of the algorithm if cluster assignments do not change between subsequent iterations. We limited ourselves to a moderate number of iterations keeping in mind the quadratic complexity of the EventMiner algorithm. For each system, we picked top-5 clusters ranked by their scores.

After obtaining the top-most clusters, we next take the most frequent keywords in each cluster and concatenate them into one sentence and then each of the sentences is concatenated into a *system-generated* summary. We compare this *system* summary with the *model* or ground truth summary from the corresponding *Wikipedia* entry of the issued keyword query. For comparing the summaries, we use the Rouge-1 score which tests the overlap of uni-grams between the two summaries and provides us with *precision*, *recall* and F_1 scores *averaged* over all the queries in that category. Note that the order of the words appearing in each summary has no consequence on the scores. The Rouge software package[5] was utilized for this purpose. We computed these scores by stemming all the words and by removing all stopwords in all the summaries. Further, 1000 samples were considered for its bootstrap re-sampling. The results reported are within 95% confidence intervals.

Results for the three different categories of the history-oriented queries are shown in Table 2. The results for the history-oriented queries concerning events show that considering EventMiner algorithm with similarity-based models for semantic annotations increases precision as compared to the baseline method considering only surface level similarity. This, however, comes at marginal cost of decrease in recall. Taking into account time and geographical model that considers proximity increases the precision of the events identified by the EventMiner algorithm again at small decrease in recall. This trend is replicated in other history-oriented queries concerning wars and entities.

We additionally describe some anecdotal results produced by EventMiner in the Appendix.

Category	System	Rouge-1		
		Recall	*Precision*	F_1
Event	EMBaseline	0.31	0.24	0.17
	EMSimilar	0.29	0.26	0.17
	EMProximity	0.25	0.28	0.18
War	EMBaseline	0.15	0.31	0.17
	EMSimilar	0.15	0.32	0.17
	EMProximity	0.11	0.36	0.15
Entity	EMBaseline	0.13	0.49	0.16
	EMSimilar	0.12	0.50	0.15
	EMProximity	0.10	0.52	0.15

Table 2: Rouge-1 scores for various systems which are grouped by different categories of history-oriented queries. Precision, recall and F_1 scores are averaged for all queries in that category for each system. Across query categories, we can clearly see an increase in the values of precision as we consider more advanced models for time and space that incorporate proximity.

5.3 Discussion

In conducting this research, we faced several pitfalls which we aim to tackle in future.

Quality of Annotations. We noticed that empirically the quality of annotations for temporal expressions or for geographic locations is of significantly better quality than for named entities. Thus, there is a need of restricting the analysis to only high precision annotations.

Cluster Coherence. Considering a joint similarity between time and geographic locations can cause cluster coherence to decrease. This might arise due to the fact that some keyword queries may have large temporal ambiguity but little or no geographic ambiguity and vice-versa.

Dependencies between Annotations. Clearly assuming an independence assumption between the annotations has not helped us in recalling more events. Our model thus needs to incorporate this aspect.

Scalability. Worst-case time complexity of the EventMiner algorithm is quadratic; this is not desirable if analysis is required for large number of documents. To tackle this, one potential solution is to use hierarchical Dirichlet process mixture models [32, 33].

Evaluation Metrics. Our evaluation metric was highly objective and based on overlap of n-grams computed from text. However, Rouge metric can additionally be modified to take into account similarity between summaries in terms of time, geography and named entities in an ontology. Another avenue to explore will be to have subjective *crowd-sourced* based evaluation for the identified events.

[5] http://www.berouge.com/Pages/default.aspx

6. RELATED WORK

In this section, we describe prior models and work with respect to our problem setting.

Related Models. Prior work [23, 37] are examples of clustering algorithms that compute document similarity based on text and contextual temporal expressions. Both have leveraged the framework of Dirichlet process mixture (DPM) models. Zhu et al. [37] present a time-sensitive Dirichlet process mixture model. The authors consider the use case of organizing an email inbox by using the arrival time of these emails. For computing similarity between different instances, they utilize an exponential decay function which gives more weight to recent emails. Building on this work, Qamra et al. [23] develop a *content-community-time* model that tries to identify similar blogs in a community of blogosphere. While utilizing the framework of DPM models, we expand on the notion of temporal similarity by taking into account uncertainty and proximity. Additionally, we incorporate similarity along the geographic dimension and also similarity between named entities in an ontology.

Temporal Information Retrieval is a sub-field of IR with special emphasis on temporal information present in documents in forms of *metadata*, e.g., publication dates or *temporal expressions* [7]. Anchoring documents or queries in time is an integral part of a time-sensitive search engine. Jatowt et al. [15] address the problem of estimating the time period which the document focuses on. They do this by constructing a weighted undirected graph which captures the associations between terms and time. Gupta and Berberich [9] address the problem of providing interesting time intervals to a keyword query by using temporal expressions in pseudo-relevant document set. Building on this they can also classify keyword queries to determine if its: *i.* (a)temporal *ii.* temporally unambiguous *iii.* temporal ambiguity at different granularities (e.g., year, month, or day) *iv.* temporally (a)periodic [10].

Timeline Generation. One of the initial works for *automatically generating timelines* was by Swan and Allan [30]. For this their model utilized the *relative frequency* of important features such as named entities and noun phrases. The method involved calculating the frequency of the features at different time points and capturing the significance of its frequency by computing χ^2 statistic. One of the more recent works involves exploring documents via timelines [3]. In their approach the authors specifically pay attention to the temporal expressions in the documents. Temporal expressions are used for constructing *temporal document profiles*. These profiles are subsequently used for clustering and re-ranking of documents.

Event Detection. A large body of work exists in analyzing different kinds of semantic annotations in isolation. However, we have tried to address the interplay between different kinds of semantic annotations, which in the past has received markedly little attention.

The *Topic Detection and Tracking* (TDT) [2] was the first initiative in the direction of modeling and detecting events. The task focused on organizing an incoming stream of text first into a broad category of *topic*, built from *stories* which comprised of *events*. However, the approaches published in [2] at that time did not have access to scalable and accurate semantic annotators. In light of these new technologies, several works have leveraged these semantic annotations to find events [1, 17] and also predict future events [16, 24].

Kuzey et al. [17] most recently addressed the task of deriving events from a news corpus to use them for ontology population. This is done by constructing a *multi-view attribute* graph that incorporates features such as: textual content of document, publication date of documents, named entities (e.g., people, organization, and location) in documents with their semantic types (e.g., protest, hurricane etc.). They then present algorithms that *distill* events from this graph. Another recent publication by Abujabal and Berberich [1] mines events from semantically annotated corpora by utilizing frequent itemset mining. Both approaches disregard any special treatment for time and geographic location; which has been adequately addressed in our work.

Focusing specifically on predicting the future, Radinsky et al. [24] developed the *Pundit* algorithm. Events are modeled as a tuple of state, actor, objects, instrument, location, and time. It predicts future events by performing *hierarchical agglomerative clustering* on events. The similarity between events is computed by using distances in a semantic network. The aim is to derive future events via causality in events that have already occurred. Utilizing only temporal data, Jatowt and Yeung [16] also present a model-based clustering algorithm for predicting future events. They capture the inherent uncertainty in temporal expressions by modeling them as probability distributions. The model-based clustering subsequently derives the similarity between these distributions using the Kullback-Leibler (KL) divergence.

Paying special attention to how events are covered in historical document collections, Yeung and Jatowt [19] study topics as they change with time. They utilize the *Google News Archive* for the time period from 1990 to 2010 for thirty two different countries. For analysis, they extract temporal expressions and also topics by using topic models such as *Latent Dirichlet Allocation*. They study the distributions of various kinds of temporal expressions that specifically refer to the past. Furthermore, they then look at how topics change over time, what *triggered* the re-collection of past events, how the events were forgotten over time, and how countries are similar when comparing their topic distribution over time.

Event-Centric Search. Using semantic annotations for clustering documents and consequently using them as means for document exploration has been addressed in works [25, 26, 28]. All the approaches however do not incorporate ontological named entities in form of person or organization. Strötgen and Gertz [26] present a method for deriving events by using time and geographic locations in text to identify events. An event in their model is a co-occurrence of a temporal and geographic expression. Building on this notion of event they provide a unique framework for querying by developing a multi-dimensional query language based on the extended Backus-Norm-Form (EBNF) language. Another work by Strötgen and Gertz [28] tries to re-rank documents for a given query by computing similarity and proximity with respect to text, time, and geography. Samet et al. [25] discuss *NewsStand*, a system which allows users to find news anchored by its location on a map. *NewsStand* does this by detecting and resolving toponyms. The method predominantly involves the use of a streaming clustering algorithm. Some of the features used for ranking the clusters are the size of the clusters, number of news sources, cluster's rate of propagation, and its timestamp.

7. CONCLUSION

In this article we presented EventMiner, an algorithm that clusters sentences in a semantically annotated corpus to identify important events. Our proposed method is based on the framework of Dirichlet process mixture models. We adapted this framework to incorporate similarity along time that leverages uncertainty and proximity in temporal expressions. It also considers similarity and proximity between geographic expressions. Finally, it also accounts for similarity between named entities in an ontology. We tested our method on a collection of history-oriented queries and their corresponding Wikipedia pages to show that considering proximity between temporal and geographic dimension as well as similarity between named entities in an ontology can recall accurate events.

8. REFERENCES

[1] A. Abujabal and K. Berberich. Important events in the past, present, and future. *WWW 2015-Companion Volume*.

[2] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] O. Alonso et al. Clustering and exploring search results using timeline constructions. *CIKM 2009*.

[4] C. E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *Ann. Statist.*, 2(6):1152–1174, 11 1974.

[5] K. Berberich et al. A language modeling approach for temporal information needs. *ECIR 2010*.

[6] D. Blackwell and J. B. MacQueen. Ferguson distributions via polya urn schemes. *Ann. Statist.*, 1(2):353–355, 03 1973.

[7] R. Campos et al. Survey of temporal information retrieval and related applications. *ACM Computing Survey*, 47(2):15:1–15:41, 2014.

[8] A. X. Chang and C. D. Manning. SUTIME: A library for recognizing and normalizing time expressions. *LREC 2012*.

[9] D. Gupta and K. Berberich. Identifying time intervals of interest to queries. *CIKM 2014*.

[10] D. Gupta and K. Berberich. Temporal query classification at different granularities. *SPIRE 2015*.

[11] D. Gupta and K. Berberich. Diversifying search results using time. Research Report MPI-I-2016-5-001, 2016.

[12] D. Gupta and K. Berberich. A probabilistic framework for time-sensitive search. *NTCIR-12 2016*.

[13] W. Härdle et al. *Nonparametric and semiparametric models*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[14] J. Hoffart et al. Robust disambiguation of named entities in text. *EMNLP 2011*.

[15] A. Jatowt et al. Estimating document focus time. *CIKM 2013*.

[16] A. Jatowt and C. Man Au Yeung. Extracting collective expectations about the future from large text collections. *CIKM 2011*.

[17] E. Kuzey et al. A fresh look on knowledge bases: Distilling named events from news. *CIKM 2014*.

[18] D. J. C. MacKay and L. C. B. Peto. A hierarchical dirichlet language model. *Natural Language Engineering*, 1:289–308, 9 1995.

[19] C. Man Au Yeung and A. Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. *CIKM 2011*.

[20] P. P. Mazur and R. Dale. Wikiwars: A new corpus for research on temporal expressions. *EMNLP 2010*.

[21] D. Metzler et al. Improving search relevance for implicitly temporal queries. *SIGIR 2009*.

[22] S. Nunes, et al. Use of temporal expressions in web search. *ECIR 2008*.

[23] A. Qamra, et al. Mining blog stories using community based and temporal clustering. *CIKM 2006*.

[24] K. Radinsky et al. Learning causality for news events prediction. *WWW 2012*.

[25] H. Samet et al. Reading news with maps by exploiting spatial synonyms. *Commun. ACM*, 57(10):64–77, 2014.

[26] J. Strötgen and M. Gertz. Event-centric search and exploration in document collections. *JCDL 2012*.

[27] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[28] J. Strötgen and M. Gertz. Proximity2-aware ranking for textual, temporal, and geographic queries. *CIKM 2013*.

[29] F. M. Suchanek et al. Yago: A large ontology from wikipedia and wordnet. *Web Semantics*, 6(3):203–217, 2008.

[30] R. C. Swan and J. Allan. Automatic generation of overview timelines. *SIGIR 2000*.

[31] Y. W. Teh. Dirichlet processes. *Encyclopedia of Machine Learning*. Springer, 2010.

[32] Y. W. Teh et al. Sharing clusters among related groups: Hierarchical Dirichlet processes. *Advances in Neural Information Processing Systems*, volume 17, 2005.

[33] Y. W. Teh et al. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[34] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *AAAI Workshop on Wikipedia and AI*, 2008.

[35] C. Y. Lin. Rouge: a package for automatic evaluation of summaries. *ACL 2004*.

[36] A. Guttman. R-Trees: A Dynamic index structure for spatial searching. *SIGMOD 1984*.

[37] X. Zhu et al. Time-sensitive dirichlet process mixture models. Technical report, DTIC Document, 2005.

[38] N. Kanhabua, et al. Temporal information retrieval. *Foundations and Trends in Information Retrieval*, 9(2):91–208, 2015.

[39] L. Hu et al. TSDPMM: Incorporating Prior Topic Knowledge into Dirichlet Process Mixture Models for Text Clustering. *EMNLP 2015*.

[40] V. Zhang et al. Geomodification in query rewriting. *GIR 2006*.

APPENDIX

A. ANECDOTAL RESULTS

Next we discuss some anecdotal results for a few history-oriented queries, namely: george w bush, bill clinton, ronald reagan, ryan white, deng xiaoping, pope john paul ii, stephen hawking, us presidential elections, soviet afghanistan war, lord of the rings movie, iraq war, and iraq iran war.

Each result shows the most coherent and representative cluster with its ten most frequent keywords, geographic locations and time intervals that appear in that cluster. The results were obtained by executing the EMProximity system with 150 iterations of Gibbs Sampling and rest settings same as used for the experimental setup. The captions accompanying the figures elaborate on the event depicted.

[6] https://en.wikipedia.org/wiki/George_W._Bush
[7] https://en.wikipedia.org/wiki/Bill_Clinton
[8] https://en.wikipedia.org/wiki/Ronald_Reagan
[9] https://en.wikipedia.org/wiki/Ryan_White
[10] https://en.wikipedia.org/wiki/Deng_Xiaoping
[11] https://en.wikipedia.org/wiki/Pope_John_Paul_II
[12] https://en.wikipedia.org/wiki/A_Brief_History_of_Time
[13] https://en.wikipedia.org/wiki/Soviet-Afghan_War
[14] https://en.wikipedia.org/wiki/The_Lord_of_the_Rings_(film_series)
[15] https://en.wikipedia.org/wiki/Iraq_War
[16] https://en.wikipedia.org/wiki/Iran-Iraq_War

Keywords	[bush] [wife] [campaign] [george] [washington] [marvin] [gore] [al] [st] [louis]
Time	[12-Apr-2000 , 12-Apr-2000] [04-Aug-2000 , 04-Aug-2000] [01-Jan-2000 , 31-Dec-2000]
Locations	none
Entities	[YAGO:George_W._Bush] [YAGO:Republican_Party_(United_States)] [YAGO:Republican_National_Convention]

Table 3: An event cluster for query george w. bush. The event identified is that of the inaugural presidential campaign of George W. Bush, whose political affiliation was to the Republican Party.[6]

Keywords	[clinton] [bill] [top] [address] [news] [page] [africa] [front] [african] [home] [hillary]
Time	[01-Jan-1999 , 01-Jan-1999] [23-Aug-1998 , 23-Aug-1998] [03-Apr-1998 , 03-Apr-1998] [01-Jan-1999 , 31-Dec-1999]
Locations	[YAGO:Africa] [YAGO:White_House] [YAGO:United_States]
Entities	[YAGO:Bill_Clinton] [YAGO:Monica_Lewinsky] [YAGO:Hillary_Rodham_Clinton] [YAGO:White_House] [YAGO:Africa] [YAGO:United_States]

Table 4: An event cluster for query bill clinton. The event identified Bill Clinton's impeachment from presidency due to the affair with Monica Lewinsky in 1999.[7]

Keywords	[reagan] [ronald] [legacy] [opinion] [top] [government] [social] [politics] [corrections] [wilson] [attack]
Time	[27-Jun-2004 , 27-Jun-2004] [01-Jan-2004 , 01-Jan-2004] [07-Jun-2004 , 07-Jun-2004] [01-Jan-1911 , 01-Jan-1911] [11-Jun-2004 , 11-Jun-2004] [14-Jan-1993 , 14-Jan-1993]
Locations	[YAGO:Palo_Alto,_California] [YAGO:California] [YAGO:United_States]
Entities	[YAGO:Ronald_Reagan] [YAGO:California] [YAGO:Nancy_Reagan] [YAGO:United_States] [YAGO:Palo_Alto,_California] [YAGO:Culture_of_the_United_States] [YAGO:Dick_Cheney] [YAGO:Ron_Reagan]

Table 5: An event cluster for query ronald reagan. Ronald Reagan passed away on June 5, 2004 in California.[8] *The cluster reports his funeral which took place on June 11, 2004.*[8]

Keywords	[ryan] [school] [white] [senior] [friends] [mother] [kokomo] [edward] [president] [riley] [attendance] [family]
Time	[01-Jan-1984 , 01-Jan-1984] [01-Jan-1199 , 31-Dec-1199]
Locations	[YAGO:New_York] [YAGO:Kingston,_New_York] [YAGO:Cicero,_Illinois] [YAGO:Indianapolis] [YAGO:Kokomo,_Indiana]
Entities	[YAGO:Megan] [YAGO:Saint_Joseph] [YAGO:Edward_VI_of_England] [YAGO:New_York] [YAGO:Matt_Ryan] [YAGO:Ronald_Reagan] [YAGO:Nancy_Reagan] [YAGO:Hamilton_Heights_School_Corporation] [YAGO:Cicero,_Illinois] [YAGO:Indianapolis] [YAGO:Kokomo,_Indiana] [YAGO:Taco_Bell] [YAGO:Kelly_Ryan] [YAGO:Ryan_White] [YAGO:Kingston,_New_York] [YAGO:Kelly_Osbourne]

Table 6: An event cluster for query ryan white. It depicts the event when Ryan White was not allowed to attend school due to health concerns related to his HIV/AIDS infection.[9] *The time interval [01-Jan-1199, 31-Dec-1199] is mined due to erroneous annotation by the temporal annotator.*

Keywords	[top] [china] [deng] [lee] [news] [li] [asia] [world] [government] [nicholas]
Time	[17-Jun-1989 , 17-Jun-1989] [03-Jan-1996 , 03-Jan-1996] [01-Jan-1970 , 31-Dec-1970] [23-Jul-1989 , 23-Jul-1989]
Locations	[YAGO:China] [YAGO:Australia] [YAGO:New_York_City] [YAGO:Ithaca,_New_York]
Entities	[YAGO:China] [YAGO:Australia] [YAGO:Fang_Lizhi] [YAGO:New_York_City] [YAGO:Tiananmen_Square_protests_of_1989] [YAGO:Deng_Xiaoping] [YAGO:Overseas_Chinese] [YAGO:Ithaca,_New_York]

Table 7: An event cluster for query deng xiaoping. It reports the Tiananmen Square protests of 1989; in which Fang Lizhi and Deng Xiaoping were key named entities.[10]

Keywords	[pope] [opinion] [savior] [cahill] [thomas] [paul] [john] [ii] [editor] [top] [catholic] [april] [saddened]
Time	[08-Apr-2005 , 08-Apr-2005] [02-Apr-2005 , 02-Apr-2005] [05-Apr-2005 , 05-Apr-2005] [03-Apr-2005 , 03-Apr-2005]
Locations	[YAGO:Florida] [YAGO:West_Palm_Beach,_Florida]
Entities	[YAGO:Pope_John_Paul_II] [YAGO:Thomas_Cahill] [YAGO:Kingdom_of_Italy] [YAGO:Camillo_Ruini] [YAGO:Catholic_Church] [YAGO:Judaism] [YAGO:Florida] [YAGO:West_Palm_Beach,_Florida]

Table 8: An event cluster for query pope john paul ii. The event described is that of his death on April 2, 2005.[11]

Keywords	[dr] [black] [hawking] [hole] [gilliam] [physicist] [time] [york] [caltech] [mr]
Time	[01-Nov-1998 , 30-Nov-1998] [01-Jan-1999 , 31-Dec-1999] [01-Jan-1900 , 01-Jan-1900] [03-Apr-1988 , 03-Apr-1988] [01-Jan-1988 , 31-Dec-1988]
Locations	[YAGO:University_of_Cambridge] [YAGO:Wildwood,_New_Jersey] [YAGO:Arizona]
Entities	[YAGO:Larry_Doyle_(writer)] [YAGO:University_of_Cambridge] [YAGO:Leonard_Susskind] [YAGO:Robert_Redford] [YAGO:William_Morris] [YAGO:A_Brief_History_of_Time] [YAGO:Wildwood,_New_Jersey] [YAGO:Arizona] [YAGO:Associated_Press] [YAGO:Random_House] [YAGO:Lou_Gehrig]

Table 9: An event cluster for query stephen hawking*. The cluster shows the dates on which first edition of his book "A Brief History of Time" was released — 1988 and tenth anniversary edition of the book was released — 1998.*[12]

Keywords	[presidential] [elections] [2000] [election] [government] [quest] [gore] [al] [pres] [vice] [politics] [campaign]
Time	[01-Jan-2000 , 01-Jan-2000] [01-Jan-2000 , 31-Dec-2000]
Locations	[YAGO:United_States]
Entities	[YAGO:United_States] [YAGO:Al_Gore]

Table 10: An event cluster for query us presidential elections*. The cluster points to the US Presidential Elections in 2000 for which Al Gore ran as vice president.*

Keywords	[soviet] [afghanistan] [war] [military] [beginning] [party] [forces] [union] [exhibition] [mixed]
Time	[01-Jan-1938 , 01-Jan-1938] [01-Jan-1980 , 01-Jan-1980] [29-Apr-1988 , 29-Apr-1988] [01-Jan-1979 , 01-Jan-1979] [01-Apr-1988 , 01-Apr-1988] [29-Jul-1987 , 29-Jul-1987] [01-Jan-1950 , 01-Jan-1950]
Locations	[YAGO:Soviet_Union] [YAGO:Afghanistan] [YAGO:Moscow] [YAGO:Kabul] [YAGO:United_States]
Entities	[YAGO:Soviet_Union] [YAGO:Afghanistan] [YAGO:Mohammad_Najibullah] [YAGO:Moscow] [YAGO:Bosniaks] [YAGO:Kabul] [YAGO:United_States]

Table 11: An event cluster for query soviet afghanistan war*. It depicts the Soviet-Afghanistan conflict that lasted from 1979 to 1989.*[13]

Keywords	[lord] [rings] [top] [movie] [motion] [opinion] [pictures] [article] [elvis] [jackson] [trilogy] [movies]
Time	[15-Dec-2002 , 15-Dec-2002] [01-Jan-1987 , 01-Jan-1987] [25-Jan-2004 , 25-Jan-2004] [12-Nov-2002 , 12-Nov-2002] [01-Jan-2003 , 31-Dec-2003] [01-Jan-1982 , 01-Jan-1982] [11-Jan-2004 , 11-Jan-2004] [28-Dec-2002 , 29-Dec-2002] [07-Sep-2003 , 07-Sep-2003] [01-Dec-2003 , 31-Dec-2003]
Locations	[YAGO:Weldon,_Northamptonshire] [YAGO:Wellington]
Entities	[YAGO:J._R._R._Tolkien] [YAGO:Weldon,_Northamptonshire] [YAGO:Wellington] [YAGO:Carol_Ann_Lee] [YAGO:Peter_Jackson]

Table 12: An event cluster for query lord of the rings movie*. It captures the location where the movie was shot — Wellington and the author of the book on which the movie is based on — J. R. R. Tolkien.*[14]

Keywords	[iraq] [states] [united] [war] [opinion] [top] [international] [relations] [defense] [armament] [president] [time] [fearful] [david]
Time	[13-Apr-2006 , 13-Apr-2006] [15-Jun-2005 , 15-Jun-2005] [16-Jul-2003 , 16-Jul-2003] [16-Oct-2003 , 16-Oct-2003] [30-Jun-2005 , 30-Jun-2005]
Locations	[YAGO:New_York_City] [YAGO:Port_Washington,_Wisconsin] [YAGO:Radcliff,_Kentucky] [YAGO:Iraq] [YAGO:United_States]
Entities	[YAGO:Iraq] [YAGO:United_States_Army] [YAGO:Donald_Rumsfeld] [YAGO:United_States_Department_of_Defense] [YAGO:George_W._Bush] [YAGO:Jim_Folsom] [YAGO:New_York_City] [YAGO:Port_Washington,_Wisconsin] [YAGO:Radcliff,_Kentucky] [YAGO:United_States]

Table 13: An event cluster for query iraq war*. The cluster shows the start of Iraq War in 2003.*[15]

Keywords	[iraq] [iran] [war] [oil] [international] [top] [faw] [port] [east] [world] [delegate] [rafsanjani]
Time	[01-Mar-1986 , 31-Mar-1986] [01-Sep-1980 , 01-Sep-1980] [01-Sep-1980 , 30-Sep-1980] [01-Jan-1970 , 31-Dec-1970] [01-Jan-1980 , 01-Jan-1980] [23-Sep-2003 , 23-Sep-2003] [25-Jan-1991 , 25-Jan-1991] [01-Aug-1988 , 31-Aug-1988] [17-Mar-2006 , 17-Mar-2006] [01-Jan-1000 , 01-Jan-1000] [01-Jan-1988 , 31-Dec-1988] [02-Oct-2003 , 02-Oct-2003]
Locations	[YAGO:Iran] [YAGO:Iraq] [YAGO:Geneva]
Entities	[YAGO:Iran] [YAGO:Iraq] [YAGO:United_Nations] [YAGO:Akbar_Hashemi_Rafsanjani] [YAGO:Iranian_peoples] [YAGO:Gulf_War] [YAGO:Geneva] [YAGO:United_Nations_Security_Council] [YAGO:Fao_Landing] [YAGO:Western_world] [YAGO:Persian_people] [YAGO:Iran-Iraq_War] [YAGO:National_Iraqi_News_Agency]

Table 14: An event cluster for query iraq iran war*. It describes the conflict between Iran and Iraq that lasted from 1980 to 1988.*[16]