WikipEvent: leveraging Wikipedia Edit History for Event Detection

Tuan Tran, Andrea Ceroni, Mihai Georgescu, Kaweh Djafari Naini, and Marco Fisichella

L3S Research Center, Appelstr. 9a, Hannover, Germany
{ttran,ceroni,georgescu,naini,fisichella}@L3S.de

Abstract. Much of existing work in information extraction assumes the static nature of relationships in fixed knowledge bases. However, in collaborative environments such as Wikipedia, information and structures are highly dynamic over time. In this work, we introduce a new method to extract complex event structures from Wikipedia. We propose a new model to represent events by engaging multiple entities, generalizable to an arbitrary language. The evolution of an event is captured effectively based on analyzing the user edits history in Wikipedia. Our work provides a foundation for a novel class of evolution-aware entity-based enrichment algorithms, and considerably increases the quality of entity accessibility and temporal retrieval for Wikipedia. We formalize this problem and introduce an efficient end-to-end platform as a solution. We conduct comprehensive experiments on a real dataset of 1.8 *million* Wikipedia articles to show the effectiveness of our proposed solution. Our results demonstrate that we are able to achieve a precision of 70% when evaluated using manually annotated data. Finally, we make a comparative analysis of our work with the well established Current Event Portal of Wikipedia and find that our system *WikipEvent* using *Co-References* method can be used in a complementary way to deliver new and more information about events.

Keywords: Event Detection, Temporal Retrieval, Wikipedia, Clustering

1 Introduction

Wikipedia is one of the largest online encyclopedias available in multiple languages. The enormous volume and the fairly reliable quality of information makes Wikipedia a popular source in several research topics. Research utilizing Wikipedia has attracted a large spectrum of interest over the past decade, including knowledge discovery and management, natural language processing, social behaviour study, information retrieval, etc. Much of existing work considers Wikipedia as a static collection, i.e. information once stored is stable or rarely changed over time. However, in practice, Wikipedia grows very rapidly (from 17 millions articles in 2011 to 30 millions articles in 2013 [1]), with new articles published and edited everyday by a large community of active contributors worldwide. This calls for an effective way to analyze and extract information from Wikipedia, with the awareness of temporal dynamics.

In this work, we address the problem of extracting from Wikipedia complex event structures, consisting of a set of entities that are connected at a given time period. We exploit the edit history in Wikipedia, which covers a full evolution of articles' content

[1] http://en.wikipedia.org/wiki/Wikipedia

over a long time period. Our method is agnostic to any language constraints, i.e. it can be applied to an arbitrary language, and it does not depend on the number of entities to be known a priori. In principle, our method can detect events from simple schema (such as the release of *new movies*) to complex ones (such as a *revolution*). Our method works naturally with the dynamics of information in Wikipedia; thus, it is able to detect several events pertaining to a number of articles, as the articles' contents change over time. In contrast to previous work in detecting events from Wikipedia [12], our model requires no training data, nor does it need the prior information about the entities establishing an event.

Detecting such dynamic relationships and associated events poses multiple interesting technical challenges. First, these relationships do not conform to any pre-existing schema and therefore can not be discovered by leveraging language patterns as in previous works on static relationship extraction. Second, the underlying events often have a flexible timeline that is difficult to know a priori, e.g., one event can last for a short time (e.g. over a day or week), while others could last over several weeks or months. Third, the entities display a great deal of flexibility in their participation in the underlying events, mainly reflected in the number of participants (some events can involve two entities while others are among several entities [8]). Fourth, as a real-life event happens, the Web community mobilizes itself to report that. Some information generated in a particular time period will no longer be available in a future version of the articles of the entities involved in the event. Thus, it is important to provide users the possibility to access historical information, giving a comprehensive evolution-aware entity-based view.

In this work, we make the following contributions:

1. Presentation of a general model which is agnostic to linguistic constraints, thus it can be applied to Social Media in different languages.
2. Establishment of new methodology for detecting events based on explicit relationships identification.
3. Introduction of the temporal aspect as a fundamental dimension to enrich content with semantic information via historical user edits.

The rest of this paper is organized as follows. We discuss related work in Section 2. In Section 3, we introduce and formalize the problem of dynamic relationships and event discovery, and present a pipeline framework for our approach. Sections 4 and 5 describe details of our approach. In Section 6, we evaluate our approach and demonstrate its effectiveness. Additionally, in this Section we conduct an extensive comparative analysis of our work with the well established Current Event Portal of Wikipedia [2] by analyzing the events described manually by Wikipedia users versus the events detected by our model. Finally, we provide our conclusions in Section 7.

2 Related Work

2.1 Temporal Information Retrieval and Event Detection

In the area of temporal information retrieval, previous works (e.g. [2, 9, 24]) leverage time dimension in different ranking models to improve the retrieval effectiveness for temporal queries, often with temporal information preprocessing. In our work, we circumvent the need of indexing or extracting temporal information from Web Archives by using Wikipedia. The link structures in Wikipedia has been shown to be a good

[2] `http://en.wikipedia.org/wiki/Portal:Current_events`

indicator for the historical influence of people [23]. Here we propose to exploit the time dimension in articles' revision to identify historical events.

The problem of identifying events has been examined using web articles as part of a broader initiative named topic detection and tracking [1]. A rich body of work has been devoted to identifying events as a landscape view of a web collection, which answers questions such as "What Happened?", "What is New?". Related to our work in this area are two approaches, article-based [1, 11] and entity-based [14]. In the article-based approach, events are detected by clustering articles based on semantics and timestamps. In the entity-based approach, temporal distributions of entities in articles are used to model events. In our work, we identify events by dynamic connections among entities that are coupled at a given time period. We formalize this problem and introduce an end-to-end pipeline as a solution.

The approach of using dynamic connections to model events has been recently proposed in [8]. The authors exploited proprietary query logs to measure the temporal interest towards an entity, using word matching as major techniques. The connections between two entities are cast into whether their query histories exhibit peaks at the same time, arguing that this is the proxy of the concurrent interests to both entities. The empirical experiments, while showing relative improvement in event discovery, were obviously affected by the performance of the word matching techniques, as well as the burst correlation algorithms. In our work, we provide a systematic framework to investigate the different types of dynamic connections of entities in many directions. In addition, the choice of Wikipedia eliminates the effects of ambiguity onto event detection performance. We also utilize high-quality ontologies such as YAGO2 [16] to support different entity classes and event domains.

2.2 Mining from Wikipedia

There has been a large amount of research done on Wikipedia. A survey [19] categorizes and presents the different areas of research to which Wikipedia is relevant. Related to our work is a rich body of work on measuring semantic relatedness of words and entities, from making use of expert-curated taxonomy such as Wordnet [4] exploiting the structure of Wikipedia to compute at larger scale [13]. Wikipedia has been also used as a rich source to measure the semantic relatedness between entities (such as people, songs, organizations, etc.), based on the inter-linking structures of Wikipedia articles [22, 20], or on the phrasal overlaps extracted from Web articles [15]. Our work distinguishes itself from existing work in the sense that it emphasizes the temporal dynamics of entities on Wikipedia, i.e. two entities with low correlation can be very relevant to one another within a particular time span, driven by an underlying event (for example, *Barack Obama* and *Iraq*). We adapt the existing measurements to our own metrics accordingly.

There are also recent attempts in extracting and summarizing historical events from Wikipedia articles' text [18]. Analyzing the trends in article view statistics, instead of article edits, the authors of [7] identify concepts with increased popularity for a given time period, and [21] proposes a system to visualize and explore the temporal correlations between different entities. A machine learning based framework for identifying and presenting event-related information from the Wikipedia edits is proposed [12], but only for individual entities. Other work proposed building a set of related entities to build the events of one entity as reference [25]. Contrast with previous work, we exploit the increased Wikipedia editing activity in the proximity of news events, and we use the edit history to identify events and entities showing similar

behaviour that might be affected by the same event, in order to extract relationships. To the best of our knowledge, we are the first to propose detecting complex events of multiple entities from Wikipedia edit history in an unsupervised fashion.

3 Approach

3.1 Overview

In this work, we aim to detect events from Wikipedia users' edit history. Existing works model events by actions, for instance, through an RDF triple of subject-predicate-object [18]. In our work, we model an event indirectly through its participating entities: one event consists of a set of entities that are connected at a given time period. For example, the event "83rd Academy Best Actor Awards" on January 25, 2011, can be described by its nominees and winners "Colin Firth", "Jeff Bridges", "James Franco", etc. This way of representing events has a benefit of being agnostic to linguistic constraints of a certain language. On the other hand, it is crucial to define the notion of entity relationships to govern an event. Such relationships must capture well the temporal dynamics of entities in Wikipedia, where information are constantly added or updated over time.

Entity Relationships. We adopt two strategies to identify entity relationships. The *Explicit Relationship Identification* uses links between Wikipedia articles to establish the relationship between their corresponding entities. The intuition behind is that each link newly added or updated in each article revision indicates explicitly a tie between the source and destination entities. For example, during the Egypt Revolution 2011, the Wikipedia article "Hosni_Mubarak" admits many revisions published. In many of them, the link to the article "Tahrir_Square" is added or refined several times. This reveals a strong relationship between the two corresponding entities with respect to the revolution. We detail methods of this strategy in Sections 4.1 and 5.1. In the *Implicit Relationships Identification* we adapt the approach presented in [8] to our domain, in order to define the entity relationships through burst patterns. This is also in line with existing work, which suggests that Wikipedia article view or article edit statistics follow bursty patterns, with spikes driven by real-world events of the entities [7, 12]. However, to avoid the coincidence of two independent entities which burst around the same time period by chance, we further impose that the entities must share sufficient textual or structure similarities during time period of study. Besides the point-wise mutual information (PMI), which was used in [8], we propose other classes of similarity measures to estimate the confidence of each implicit relationship. This is discussed in more detail in Section 4.2.

Event Detection. Having defined entity relationships, we detect events by building groups of highly related entities, each representing an event. We cast this problem to the connected components extraction from the graph, with the nodes corresponding to Wikipedia entities and edges corresponding to entity relationships. One subtle problem in identifying such components is that the graph is highly dynamic, i.e. edges change as entity relationships evolve over time, new relationships can be established in a given time and dissipate later, when the tie between two entities gets weaker within its respective revisions. For instance, two entities "Barack Obama" and "Mitt Romney" are highly related during the US presidential election 2012, but they rarely correlate long before and after the event. In this work, we propose an adaptive algorithm that handles this temporal dimension, which will be detailed in Section 5.

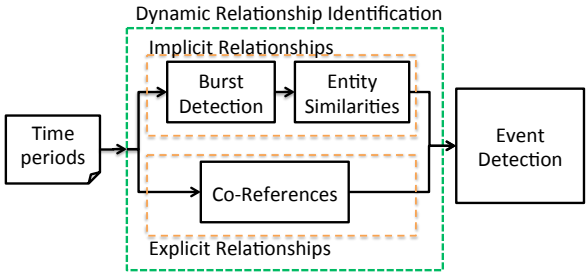


Fig. 1: Architecture for identifying events and relationships between entities

3.2 Problem Formalization

Data Model. In our model, time is represented as a sequence of discrete points, each corresponding to a day and indexed by $i = 1, 2, 3, \ldots$ Let E denote the entity collection derived from Wikipedia, where each entity e is associated to a Wikipedia article, and let $\boldsymbol{\tau}$ be the set of time points that we consider. At a given time point i, e is represented as a textual document $d_e^{(i)}$, which is the revision of the article at a latest timestamp t that was before i. Given such assumptions, we further define the *edit* of e at the time point i as $m_e^{(i)} := d_e^{(i)} - d_e^{(i-1)}$ and the *edit volume* $v_e^{(i)}$ as the number of revisions between two time points $i-1$ and i.
Dynamic Relationships. A dynamic relationship is a tuple $r := (e_1, e_2, i)$, where $e_1, e_2 \in E$ are the entities for which r holds, and $i \in \boldsymbol{\tau}$ is the time when r is valid. Dynamic relationships can be of two types: *explicit* and *implicit*. They are identified according to different strategies (Section 4).
Events. We define an *event* v as a tuple $v := (E_v, \boldsymbol{\tau}_v)$, where $E_v \subset E$ is the *representative entity set*, i.e. entities that participated to v and contributed to its interpretation, and $\boldsymbol{\tau}_v := \{i : i \in \boldsymbol{\tau}, i_{start} \leq i \leq i_{end}\}$ is the *time period* when v occurred.
Problem statement. Given an entity set E, a time window $\boldsymbol{W} \subseteq \boldsymbol{\tau}$, detect all events $v = (E_v, \boldsymbol{\tau}_v)$ such that $E_v \subset E$ and $\boldsymbol{\tau}_v \subset \boldsymbol{\tau}$.

3.3 Workflow

The detailed workflow of our system *WikipEvent* is described in Figure 1. In short, our system consists of two steps: *Dynamic Relationship Identification* and *Event Detection*. Givena time period as input, our system detects a set of events in which the entities were involved, together with the relationships between them. Such relationships, together with the specified time period, will enable users to fully interpret the detected events (e.g., causes and effects) [3]. The first phase computes the dynamic relationships between entities, using one of the two strategies discussed in Section 4.1 (*Explicit Relationships Identification*) and in Section 4.2 (*Implicit Relationships Identification*). The *Explicit Relationships Identification* strategy uses explicit links in the articles to establish the relationship between two entities. Its intuition is that

[3] The semantic extraction between entities is not the focus of this work

each link added or modified in one revision encodes a binding between the source and destination articles, hence the entities. The adapted and extended *Implicit Relationships Identification* strategy is based on two steps. First, we use *Burst Detection* to detect salient bursts of activity in the edit history as described in Section 4.2. The output is a set of pairs of entities that have bursts during the same time point. We employ a variety of methods to measure their similarity in the *Entity Similarity* step, described in Section 4.2. The pairs of entities are aggregated using the previously computed similarities to build the co-burst graph for each individual time point.

The second phase, *Event Detection*, generates events described by representative entities and time intervals of involvement. It first builds a sequence of graphs, each one capturing the entity relationships at an individual time point. It then incrementally builds the connected components that group entities that are highly related in consecutive time points.

4 Relationship Identification

In this section, we present two strategies to create dynamic relationships, already introduced in Section 3.1.

4.1 Explicit Relationships Identification

In our established strategy, hereafter called *Co-References*, we define entity relationships as follows. For each entity e and an edit $m_e^{(i)}$ at time i, if $e_2 \in m_e^{(i)}$ then it implies that there exists a link to the Wikipedia article of the entity e_2 in the content of $m_e^{(i)}$. A relationship between e_1 and e_2 is established if we have links in both directions (from e_1's edit to e_2 and vice versa).

Definition 1. *Given two entities e_1, e_2 and a time interval $I = [i, i+\delta]$, an explicit dynamic relationship between e_1 and e_2 at time point i is a tuple $r_{exp} \in E \times E \times \tau$ such that $r_{exp}(e_1, e_2, i)$ iff $\exists j, k \in [i, i+\delta], e_1 \in m_{e_2}^{(j)}$ and $e_2 \in m_{e_1}^{(k)}$.*

Intuitively, an explicit dynamic relationship captures the mutual references between two entity edits that are made at close time points. The parameter δ accounts for the possible time delay when adding links between two entities. As an example, while the entities "Cairo" and "Hosni Mubarak" are explicitly related during the Egypt revolution in 2011, the two mutual references can be added at different (close) time points; for instance the link from the article of "Hosni Mubarak" to "Cairo" can be added first, and the inverse link can be added one day after.

4.2 Implicit Relationships Identification

We adapted to the Wikipedia domain the strategy proposed in [8]; in this adapted approach, we identify a relationship between two entities when their edit histories exhibit bursts in the same or overlapped time intervals (co-burst). A burst of an entity is the time interval in which the edit volume of the entity is significantly higher than the preceding and following volumes. More formally, given an entity e and a time window $\boldsymbol{W}$, we construct a time series $\boldsymbol{v}_e := [v_e^{(i_0)}, \ldots, v_e^{(i_f)}]$ containing the edit volume of entity e at every time point $i \in \boldsymbol{\tau}$. A burst $\boldsymbol{b}_e$ is a sequence of time points $\boldsymbol{b}_e := [i, i+1, \ldots, i+k]$ for which the edit volumes of e are *significantly* higher than the edit volumes observed in neighbouring time points: $v_e^{(i-1)} \ll v_e^{(j)}$ and $v_e^{(j)} \gg v_e^{(i+k+1)} \forall j = i, i+1, \ldots, i+k$. We say two entities e_1 and e_2 *co-burst* at time i iff $i \in \boldsymbol{b}_{e_1}$ and $i \in \boldsymbol{b}_{e_2}$.

A *co-burst* merely identifies entities that admit high volume of edits at the same time. In practice, two entities can have bursts in the same time interval, even if they are of little relevance. To remedy this, we further assume that the edits of two entities must share sufficient resemblance, which we assess through 4 different similarities. Let us denote the similarity of two entities e_1 and e_2 at time i as $S_{method}(e_1, e_2, i)$, then we have the following similarities:
1. Textual: it measures how close two entities are in a given time by comparing the content of their corresponding edits. We construct the bags of words $\boldsymbol{bw}_{e_1}^{(i)}$ and $\boldsymbol{bw}_{e_2}^{(i)}$, and use Jaccard index to measure the similarity: $S_s(e_1, e_2, i) := J(\mathbf{bw}_{e_1}^{(i)}, \mathbf{bw}_{e_2}^{(i)})$.
2. Entity: This is similar to textual similarity, but with the *bag of entities* $\boldsymbol{be}_e^{(i)}$ (entities that are linked from the edit): $S_e(e_1, e_2, i) := J(\boldsymbol{be}_{e_1}^{(i)}, \boldsymbol{be}_{e_2}^{(i)})$.
3. Ancestor: it measures how close two entities are in terms of their semantic types. For each entity, we use an ontological knowledge base where the entity is registered, and extract all its ancestors (entities that are connected to through a *subsumption* relation). Given $\boldsymbol{be}_e^{(i)}$, a *bag of ancestors* $\boldsymbol{ba}_e^{(i)}$ is filled with the ancestors of every entity in $\boldsymbol{be}_e^{(i)}$. We then measure the similarity $S_a(e_1, e_2, i)$ by Jaccard index accordingly.
4. PMI: This measures how likely two entities co-occur in the edits in all other entities. Given $i \in \boldsymbol{\tau}$ and $e_1, e_2 \in E$, we construct the graph involving all entities linking to e_1 and e_2 from all edits at i. Let $IN(e)^{(i)}$ denote the number of incoming links for e in this graph, we estimate the probability of generating e by $p(e) = \frac{IN(e)^{(i)}}{N^{(i)}}$, with $N^{(i)}$ being the total number of incoming links in the graph at time i. We then computed the link similarity as $S_{PMI}(e_1, e_2, i) := \log \frac{p(e_1, e_2)}{p(e_1)p(e_2)}$.

5 Event Detection

Having defined the dynamic entity relationships, we now aim to detect events by identifying their representative entity sets and the corresponding time period. The event detection is done via an incremental approach as follows. At each individual time point $i \in \tau$, we consolidate different relationships into one unified group, called *temporal graph*. The temporal graph can be thought of as one event's snapshot at the specified time point (Section 5.2). To accomodate the development of events from time points to time points, as well as to factor the event time period, we compare entity clusters of two adjacent temporal graphs, and incrementally merge two clusters if a certain criterion is met. The resulting merged set of entities represents the event that evolves across several continuous days, from temporal graph to temporal graph. Our event detection is detailed below. We first start with the formal definition of the temporal graph and entity clustering.

5.1 Temporal Graph and Entity Clustering

Definition 2. *A* temporal graph $G(i)$ *at time* $i \in \tau$ *is an undirected graph* (E, P)*, where* E *is an entity set and* $P = \{(e_1, e_2) | r(e_1, e_2, j)\}$ *is the set of edges defined by dynamic relationships at a time point* $j \in \boldsymbol{I} = [i, i + \delta]$.

In the above definition, the value δ reflects the lag of edit activities between different Wikipedia articles in response to one real-world event. Note that depending on the type of the dynamic relationships, we have two different types of *explicit* and *implicit* temporal graph respectively.
Explicit Temporal Graph Clustering. In an explicit temporal graph, an edge is defined by the relationship $r_{exp}(e_1, e_2, j)$ and it reflects the mutual linking structure of two Wikipedia entities within interval I. From the temporal graph, we identify the

Algorithm 1: Entity Cluster Aggregation Algorithm

Input : $E, \boldsymbol{W}, \gamma$ -cluster merging threshold, *strategy*
Output: C_{ret} as entity sets representing events
Set $C_{ret} = \emptyset$
for each $i \in |\boldsymbol{W}|$ **do**
  Construct temporal graph $G(i)$
  C_{I_i} = clusters in $G(i)$
  for each $c_k \in C_{I_i}$ **do**
    merged = False
    for each $c_j \in C_{I_{i-1}}$ **do**
      if ($strategy = explicit$ & $\frac{|c_k \cap c_j|}{|c_k \cup c_j|} \geq \gamma$) **or**
      ($strategy = implicit$ & $|c_k \cap c_j| \geq 1$) **then**
        $c_k = c_k \cup c_j$ (merge the two events)
        merged = True
      end
    end
    if not merged **then**
      $C_{ret} = C_{ret} \cup c_k$
    end
  end
end
return C_{ret}

set of maximum cliques C to form clusters of entities that are mutually co-mentioned from i to $i + \delta$. Each maximum clique $c \in C$ represents an event that occurs at i. The choice of cliques in favor of connected components in this case ensures the high coherence of the underlying events encoded in the group of entities. For example, considering three entities "Anne Hathaway", "James Franco" and and "Minute To Win It" during January 2011. The first two entities are connected by the fact that the two actors co-hosted the ceremony of the 83rd Academy Awards, while the second and third entities are connected because James Franco was at that time a co-performer in the show. This forms a connected component, but putting the three entities together reveals no obvious event.

Implicit Temporal Graph Clustering. In an implicit temporal graph, a candidate edge will be established from two entities which co-burst at a time point $j \in \boldsymbol{I} = [i, i + \delta]$. To mitigate the "co-burst by chance" (Section 4.2), we define a *maximum* similarity function:

$$S_{max}(e_1, e_2, \boldsymbol{I}) = \max_{j \in [i, i+\delta]} \{S(e_1, e_2, j)\}$$

and create an edge (e_1, e_2) iff $S_{max}(e_1, e_2, \boldsymbol{I}) \geq \theta$. Intuitively θ is the threshold value used to perform a selective pruning preserving only entity pairs with maximum similarities exceeding it. Unlike in an explicit temporal graph, here we relax the entity clustering requirements by representing the events occuring at i as the connected components.This is due to the nature of the implicit dynamic relationships, where two entities e_1 and e_2 that are not directly connected can still co-burst, through an intermediate entity e' during the interval $\boldsymbol{I}$, by following one path in the graph.

5.2 Event Identification

To identify an event, we need to form a representative entity set from a number of temporal graphs, as well as to specify the time interval in which the entity set lies in. This entails aggregating entity clusters of temporal graphs at consecutive time points. The algorithm named *Local Temporal Constraint* (*LTC*), proposed in [8], detects events in the dynamic programming fashion. At each time point i, *LTC* maintains a set of merged clusters from the beginning until i, and merges these clusters with those in the temporal graph at time $i+1$. Two clusters are merged if they share at least one edge; i.e. one dynamic relationship. However, we observe that this can end up merging a lot of clusters where entities across clusters are very loosely related. We adapt the algorithm as follows. For the implicit temporal graph, we keep merging two connected components if they share one edge (i.e. original *LTC* algorithm), while for explicit temporal graph, we only merge two clusters c_1 and c_2 if their Jaccard similarity is greater than a threshold γ. The pseudocode of our algorithm is detailed in Algorithm 1.

Complexity Let M be the maximum number of events within each event set C_I. Let n be the maximum number of relationships which can be found inspecting each event within each C_I. Let $\mathcal{T}$ be the number of intervals, then the computational cost of *LTC* is $O((\mathcal{T}-1)nM^2)$.

6 Experiments and Evaluations

In order to analyze the performances of the proposed methods, we ran experiments on the quantitative (Section 6.3) and qualitative (Section 6.4) characteristics of our extracted events. For the specific task of detecting event structures in Wikipedia, to the best of our knowledge, a comprehensive list of real-world events to be used as universal ground truth does not exist: existing resources (e.g. Wikipedia Current Event Portal, YAGO2 [16]) are limited in terms of number, complexity, and granularity of events. Moreover, since a comprehensive event repository does not exist, fairly computing recall for event detection methods is infeasible. Thus, we performed a manual evaluation as follows. Detected events were manually assessed by five evaluators who had to decide if they were corresponding to real events. In detail, for each detected event, the annotators were asked to check all involved entities and identify a real-world event by examining web-based sources (Wikipedia, official home pages, search engines, etc.), that best explained the co-occurrence of these entities in the event during the specified time period. For each set of entities, a label was assigned in order to represent a *true* or *false* event. These assessments contributed to measure the performances of our methods.

Finally, we conducted an extensive comparative analysis of our work with the well established WikiPortal by analyzing the events described manually by Wikipedia users versus the events detected by our best performing method (Section 6.5).

6.1 Dataset

To build our dataset, we used the English Wikipedia. Since Wikipedia also contains articles that do not describe entities (e.g., "List of mathematicians"), we selected Wikipedia articles corresponding to entities registered in YAGO2 and belonging to one of the following classes: person, location, artifact, or group. In total we got $1,843,665$ articles, each corresponding to one entity. We chose a time period ranging from the 18^{th} January 2011 until the 9^{th} February 2011, because it covers important real-

world events such as the Egypt Revolution, the Academy Awards, the Australian Open, etc. The choice of a relatively short time span simplifies the manual evaluation of the detected events. Since using days as time units has been shown to effectively capture the news-related events in both social media and newswire platforms [3], we used the day granularity when sampling time. We name the whole dataset, containing all the articles, as *Dataset A*. Furthermore, we created a sample set, called *Dataset B*, by selecting entities that were actively edited (more than 50 times) in our time period. The intuition behind this selection is that a large number of edits is more likely to be caused by an event. Consequently, this sample contains just $3,837$ Wikipedia articles.

6.2 Implementation Details

Entity Edits Indexing To store the whole Wikipedia edit history dump and to identify the edits, we made use of the JWPL Wikipedia Revision Toolkit [10]. JWPL solves the problem of storing the entire edit history of Wikipedia by computing and storing differences between two revisions.
Similarity To resolve the ancestors of a given entity, we employed the YAGO2 knowledge base [16], an ontology that was built from Wikipedia infoboxes and combined with Wordnet and GeoNames to obtain 10 million entities and 120 million facts. We followed facts with *subClassOf* and *typeOf* predicates to extract ancestors of entities. We limited the extraction to three levels, since we observed that going to a higher level included several extremely abstract classes (such as "Living people"). This lowered the discriminating performance of the similarity measurement.
Burst Detection and Event Detection We implemented Kleinberg's algorithm using the modified version of CShell toolkit [4]. We set the density scaling to 1.5, the transition cost to 1.0, and the default number of burst states to 3 (for more details, refer to [17]). We observed that changing parameters of the burst detection did not affect the order of performance between different event detection methods. For the dynamic relationships, we set the time lag parameter δ to 7 days and γ to 0.8, as these values yielded the most intuitive results in our experiments.

6.3 Quantitative Analysis

The goal of this section is to numerically evaluate our approach under different metrics: (i) total number of detected events, and (ii) the precision, i.e. the percentage of *true* events. For the parameter selection, note that the graph created based on the explicit strategy does not have any weights on its edges. On the contrary, the implicit strategy creates a weighted graph based on the similarities, and the temporal graph clustering depends on the threshold θ to filter out entity pairs of low maximum similarity. We varied the value of θ and noticed that lowering it resulted in a larger number of entity pairs that coalesced into a low number of large events. These events containing a large number of various entities could not have been identified as real events. Therefore, for the following experiments we used $\theta = 1$.

We evaluate approaches for the implicit relationship identification strategies as defined in Section 4.2, referred to as the following *methods*: *Textual*, *Entities*, *Ancestors*, and *PMI*, as well as for the explicit strategy as defined in Section 4.1, referred to as the *Co-References* method. The results are presented in Table 1 and Table 2. The number of events detected for the different similarity setups is presented in the third column of the tables. As expected, we detect more events in *Dataset A* as in *Dataset*

[4] http://wiki.cns.iu.edu/display/CISHELL/Burst+Detection

B, due to the higher number of entities taken into consideration. The biggest number of detected events is provided by *Co-references* in both datasets. This is attributed to the parameter-free nature of the explicit strategy, while for the implicit strategy, a portion of events are removed by a threshold. Comparing the methods used by the implicit strategy, *PMI* detects more events than any other method. This is caused by the difference in computing the entity similarity $S(e_1, e_2, t)$. *PMI* considers the sets of incoming links, that account for relevant feedback to our e_1 and e_2 from all the other entities in Wikipedia. This results in more entity pairs, and more clearly defined and coherent events, while the other implicit strategy methods tend to conglomerate most of the entities in larger but fewer events. *Textual*, *Entities*, and *Ancestors* compute $S(e_1, e_2, t)$ starting from the edited contents of two entities at a given time. A large amount of content concerning entities that are not explicitly referring to e_1 and e_2 will be taken in consideration as well, making the value of $S(e_1, e_2, t)$ lower. Therefore, using the same value for θ as for the *PMI*, produces a lower number of entity pairs, and consequently of detected events.

Table 1: Performance on Dataset A

Strategy	Method	Events	Precision
Explicit	Co-References	186	70%
Implicit	PMI	124	39%
	Ancestors	33	51%
	Entities	21	62%
	Textual	78	1%

Table 2: Performance on Dataset B

Strategy	Method	Events	Precision
Explicit	Co-References	120	80%
Implicit	PMI	80	69%
	Ancestors	18	50%
	Entities	12	60%
	Textual	15	7%

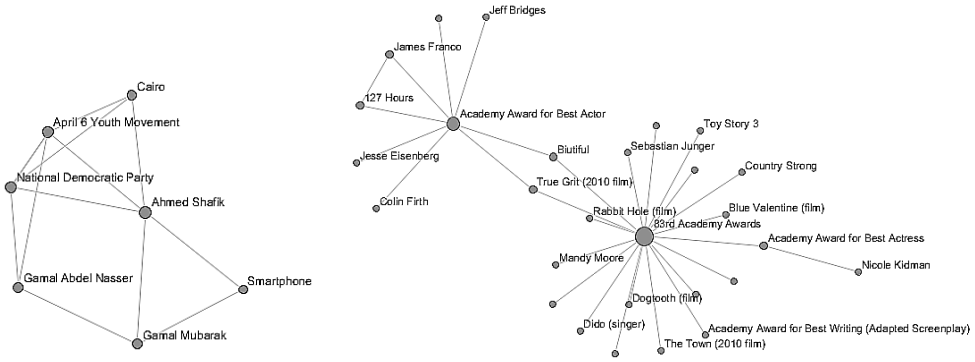
The precision of every setup, i.e. the percentage of true detected events, is summarized in the fourth column of Table 1 and Table 2. Among the implicit strategy methods, we notice a clear benefit of using similarities that take semantics into account (*Entities*, *Ancestors*, and *PMI*) over the string similarity (*Textual*). *Ancestors* performs worse than *Entities* in both datasets, showing that the addition of the ancestor entities introduces more noise instead of clarifying the relationships between the edited entities. *Entities* achieves similar performances on both datasets. *PMI* achieves better performance in *Dataset B* than the other implicit similarities since it is leveraging the structure of incoming-outgoing links between Wikipedia articles. However, *PMI* performs worse on *Dataset A* due to the higher number of inactive entities considered, introducing noisy links.

Finally, *Co-References* outperforms all the implicit strategy methods on both datasets, showing that a clear and direct reciprocal mention is stronger than similarities inferred from the text of the edit. Generally, all methods performed better or comparable on *Dataset B* in comparison to *Dataset A*. This shows that selecting only the entities that are edited more often improves the quality of our methods. Although less events are detected in *Dataset B*, more of them correspond to real life events.

6.4 Qualitative Analysis

In this section, we do a qualitative evaluation of the events identified in *Dataset A*. First, we focus on and describe some of the events detected by our best method *Co-References* (highlighted in Section 6.3). Second, we analyze some cases where our methods failed, proposing the causes.

In Table 4 we present and discuss some events identified by our best method *Co-References* matching real-world events. For each detected event, we report the entities



(a) The Friday of Anger (b) 83rd Academy Awards Nominations

Fig. 2: Example detected events

involved, the time when the event occurred, and a human description of the event extracted from web-based sources.

Moreover, we show the graph structure of two good examples of identified events: **Example I** We depict the relationships associated to the real-world event known as the Friday of Anger, in the context of the Egyptian Revolution in Figure 2a. On January 28, 2011, tens of thousands filled the streets across Egypt, to protest against the government. One of the major demonstrations took place in *Cairo*. The organization of the protests was done with the help of internet and *smartphones*, and some of the organizers were the *April 6 Youth Movement* and the *National Democratic Party*. Protesters held portraits of former president *Gamal Abdel Nasser*. The aviation minister and former chief of Air Staff *Ahmed Shafiq*, as well as *Gamal Mubarak*, were seen by the government as probable successors of Hosni Mubarak.
Example II The graph structure of another outstanding detected real world: the announcement of the nominees for the 83rd Academy Awards, on January 25, 2011, is shown in Figure 2b. We can see as a the biggest central node the *83rd Academy Awards*, and as a secondary node the *Academy Award for Best Actor*, having as connecting nodes *True Grit* and *Biutiful*, which were nominated for more categories.

In Table 3, we report some failures of our methods to identify real-world events, together with the causes that lead to such erroneous output. Depending on the method, we can notice different patterns that cause false positives. The *entity-based similarity* usually fails because of updates containing a large number of common entities that are not involved in any common event. Using the *ancestor-based similarity* can provide false events because some entities that are very similar, and share a large number of ancestors, have coincidentally concurrent edit peaks in the same period. The *PMI* fails because of similar causes: entities that share a lot of common incoming links to the entities contained in the edits done on the same day. Finally, the *Co-References* method seems to fail when the reciprocal mentions originate in relationships that are independent of any event.

6.5 Comparative Analysis

We compare events detected by our approach with events present in WikiPortal in the same time period. Users in WikiPortal publish a short description of an event in response to the occurrence of a real-world happening and annotate the entity

mentions with the corresponding Wikipedia articles. The event descriptions can also be grouped into bigger "stories" (such as Egypt revolution), or can be organized in different categories such as sports, disasters, etc.

We conducted the comparison by considering the 130 (70% of 186) true events detected with the *Co-References* method on *Dataset A*, since this is the setup that gave the largest set of true events.

Then, we collected 561 events from WikiPortal within the period of interest by considering all event descriptions inside the same story as representing a single event. We further considered only those event descriptions annotated with at least an entity contained within *Dataset A*, getting in total 505 events. In principle, these events can be detected by our method.

In order to assess the overlap between the two event sets, we classify events according to the following categories:

1. Green: The event in one set, with all its participating entities, is present in the other event set either as a single event or as multiple events.

2. Yellow: The event is partially present in the other event set, i.e. only a subset of its participating entities appears in one or more events in the other set.

3. Red: The event is not reported in the other event set.

We provide explanatory examples for each category in Table 4, along with explanations of each classification choice. We can observe 2 patterns for the events in our set belonging to the green category: (i) one event in *Co-References* is spread over different events in WikiPortal; for instance, the event regarding the candidacies for the Fianna Fail party is reported in WikiPortal through different events, each one focusing on a single candidacy; (ii) one event in *Co-References* corresponds to one event in WikiPortal. The yellow category generally covers the case where an event represents a non mentioned aspect of an event in WikiPortal. For instance, for the Australian Open tennis tournament only the men's semi-final and final matches are reported in WikiPortal, without mentioning the other matches, which are reported in *Co-References*. Similarly, the Friday of Anger (in the context of the Egyptian revolution) and the Academy Awards nominations are present in WikiPortal, but our detected events are endowed with more entities that do not appear in the portal. Finally, the red category collects those events that are not reported in WikiPortal at all, like the Royal Rumble wrestling match.

Table 3: False positive events and probable reasons why our methods failed

Entities	**Method**	**Cause**
Alexis Korner, Fleetwood Mac, Bob Brunning	Co-References	Bob Brunning was a member of the Fleetwood Mac (a British-American rock band) and Korner wrote a book about them. They were not involved in any common events in our period.
Alexa Nikolas, Ariana Grande	PMI	Two different persons, that just look alike, share some of their own incoming links, but not much more. They were not involved in a common event in our period, but their articles might have experienced unrelated edit peaks simultaneously.
Saudi Arabia national football team, Ghana national football team, Canada national soccer team	Ancestors	The entities have a lot of common ancestors, coming from the Sports domain, and all of them had peaks of activity in the same time. However, they were not involved in common events during the studied period.
Tura Satana, Barack Obama	Entities	Tura Satana died, but Barack Obama did not have any connection to her in the period under investigation, although both entities experienced edit peaks and the entities contained in the edits were similar.

We noticed that 60% of the events detected by *Co-References* are present fully or partially in the WikiPortal. For the sake of clarity, in Table 5 we present some of the events that are present in WikiPortal but our method was unable to detect, along with an explanation. The main patterns are: (i) the events involve just one entity; (ii) the events involve entities that are highly unlikely to reference each other because of their different roles in the common events.

In conclusion, WikiPortal and *Co-References* can be seen as complementary methods for event detection. While WikiPortal is user-contributed and requires human effort to curate events, our method is fully unsupervised, and can detect additional events without the human intervention.

Table 4: Co-References based extracted events and the matching real-life events, along with their date and a human description

Entities	**Date(2011)**	**Human Description**	**Category Explanation**
Category: Green			
Brian Cowen, Michel Martin, Mary Hanafin, Mary Coughlan(politician), Fianna Fáil	From January 18 to January 26	The Irish PM Brian Cowen announced his stepping down as leader of the ruling Fianna Fail party, and different candidacies for the leadership follow his decision.	The event is globally reported in WikiPortal through different daily events.
Saad Hariri, Najib Mikati	January 25	Supporters of Lebanese caretaker Saad Hariri call for a day of protests following Hezbollah's support for Najib Mikati as Prime Minister	The entities participating to the event are all mentioned in an event within WikiPortal.
Category: Yellow			
Gamal Abdel Nasser, Ahmed Shafik, Smartphone, Cairo, April 6 Youth Movement, Gamal Mubarak, National Democratic Party	January 28	In the context of the Egyptian Revolution, the Friday of Anger takes place: tens of thousands filled the streets across Egypt, to protest against the government.	Most of the entities appear in WikiPortal within different events. The entities *Gamal Mubarak* and *Gamal Abdel Nasser* do not appear.
Li Na, Kim Clijsters	January 19	Australian Open 2011 women's final: Li Na vs Kim Clijsters.	The Australian Open is mentioned two times, but always focusing on men's matches. The women's final is not reported.
James Franco, Colin Firth, Biutiful, True Grit, 83rd Academy Awards, ... (Figure 2b)	January 25	Announcement of the nominees for the 83rd Academy Awards	The event is reported in WikiPortal, but few participating entities are mentioned.
Category: Red			
Vickie Guerrero, Hornswoggle, Layla El, Dolph Ziggler, Booker T, Professional wrestling, Kane, Santino Marella	January 30	The 2011 Royal Rumble organized by WWE takes place, involving a lot of wrestlers.	The event and no one of its entities are reported in WikiPortal.
Silent Witness, Bruce Forsyth, Loose Women	January 26	In the context of the 16th National Television Awards, presented by Bruce Forsyth, Loose Women and Silent witness are nominated.	The event and no one of its entities are reported in WikiPortal.
Catwoman, The Dark Knight, Bane	January 19	Warner Bros. Pictures announced that Anne Hathaway has been cast as Catwoman and Tom Hardy as Bane in "The Dark Knight Rises".	The event and no one of its entities are reported in WikiPortal.

Table 5: Examples of events from WikiPortal that were not detected by our method, Co-References

Event Description	Date(2011)	Explanation
Apple records record profits of $6 billion as consumers consumed more of its products than was thought (BBC)	January 18	The event involves just one entity
Chinese President Hu Jintao begins a four-day state visit to the United States.	January 18	It is highly unlikely that the prominent entities have mentioned each other
Exotic birds are found to have been driven into Britain's back gardens by the extreme cold, as more than half a million people participate in the largest wildlife survey in the world	January 29	It is highly unlikely that the event attracted the attention of the Wikipedia community
Researchers report that fishing rates in the Arctic are 75 times higher than those reported by the U.N., suggesting future increased exploitation is less possible than previously thought.	February 4	It is highly unlikely that the prominent entities have mentioned each other

7 Conclusions and Future Work

In this work, we propose incorporating temporal aspect with semantic information to capture dynamic event structures. Focusing on Wikipedia, we are able to find historical information, events. Because of the specific task we consider, no annotated collections were available, thus we manually assessed the performance of our methods using a data set of 1.8 million articles. Over an extensive set of experiments we established the effectiveness of out proposed approach and investigated different strategies and methods. We have shown that an explicit relationship identification strategy performs better than an implicit one, achieving a maximum precision of 70%. We observed a further increase to 80% in precision when using only actively edited articles. We further conducted a comparison between events detected by WikipEvent, using the *Co-References* approach, with events present in WikiPortal in the same time period, highlighting that they can be seen as complementary sources of events. Future work in this direction includes using Web as a complementary resource for validating news events (e.g. [5]).

For the future work, we are investigating how a current approach using cross-references between two entities can be combined with an observed low correlation in previous time window of the same pair to improve the quality of event detection. Another direction includes adding more semantics to the event detected via text analysis, or summarizing events on Wikipedia, taking into account the evolution of engaging entities over time [6].

8 Acknowledgements

This work was partially funded by the European Commission for the FP7 projects CUbRIK and ForgetIT (under grants No. 287704 and 600826 respectively), the ERC Advanced Grant ALEXANDRIA (under grant No. 339233), and the L3S-run project WikipEvent [5]. We thank the anonymous reviewers for constructive advices about our paper and suggestions for direction of future work.

References

1. J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *ACM SIGIR*, pages 37–45, 1998.

[5] `https://www.l3s.de/en/projects/iai/~/wikipevent`

2. O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *WWW*, 2011.
3. R. Bandari, S. Asur, and B. A. Huberman. The pulse of news in social media: Forecasting popularity. In *ICWSM*, 2012.
4. A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1), 2006.
5. A. Ceroni and M. Fisichella. Towards an entity–based automatic event validation. In *ECIR*, pages 605–611. Springer, 2014.
6. A. Ceroni, M. Georgescu, U. Gadiraju, K. Djafari Naini, and M. Fisichella. Information evolution in wikipedia. In *Proceedings of the 10th International Symposium on Open Collaboration*, OpenSym '14. ACM, 2014.
7. M. Ciglan and K. Nørvåg. Wikipop: personalized event detection system based on Wikipedia page view statistics. In *CIKM*, pages 1931–1932, 2010.
8. A. Das Sarma, A. Jain, and C. Yu. Dynamic relationship and event discovery. In *WSDM*, pages 1931–1932, 2011.
9. M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *ACM SIGIR*, pages 495–504, 2011.
10. O. Ferschke, T. Zesch, and I. Gurevych. Wikipedia revision toolkit: Efficiently accessing wikipedia's edit history. In *HLT*, pages 97–102, 2011.
11. M. Fisichella, A. Stewart, K. Denecke, and W. Nejdl. Unsupervised public health event detection for epidemic intelligence. In *CIKM*, pages 1881–1884, 2010.
12. M. Georgescu, N. Kanhabua, D. Krause, W. Nejdl, and S. Siersdorfer. Extracting event-related information from article updates in wikipedia. In *ECIR*, pages 254–266, 2013.
13. S. Hassan and R. Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI*, 2011.
14. Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *ACM SIGIR*, pages 207–214, 2007.
15. J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. KORE: keyphrase overlap relatedness for entity disambiguation. In *CIKM*, pages 545–554, 2012.
16. J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Int. J*, 2012.
17. J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.
18. E. Kuzey and G. Weikum. Extraction of temporal facts and events from wikipedia. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, pages 25–32. ACM, 2012.
19. O. Medelyan, D. Milne, C. Legg, and I. H. Witten. Mining meaning from Wikipedia. *Int. J. Hum.-Comput. Stud.*, 67, 2009.
20. D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI*, 2008.
21. M.-H. Peetz, E. Meij, and M. de Rijke. Opengeist: Insight in the stream of page views on wikipedia. In *SIGIR Workshop on Time-aware Information Access*, 2012.
22. S. P. Ponzetto and M. Strube. Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Int. Res.*, 30(1), 2007.
23. Y. Takahashi, H. Ohshima, M. Yamamoto, H. Iwasaki, S. Oyama, and K. Tanaka. Evaluating significance of historical entities based on tempo-spatial impacts analysis using wikipedia link structure. In *ACM HyperText*, 2011.
24. T. Tran. Exploiting temporal topic models in social media retrieval. In *SIGIR*, 2012.
25. T. Tran, S. Elbassuoni, N. Preda, and G. Weikum. CATE: context-aware timeline for entity illustration. In *WWW*, pages 269–272. ACM, 2011.