EVIN: Building a Knowledge Base of Events

Erdal Kuzey
Max Planck Institute for Informatics
Saarbrücken, Germany
ekuzey@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

We present EVIN: a system that extracts named events from news articles, reconciles them into canonicalized events, and organizes them into semantic classes to populate a knowledge base. EVIN exploits different kinds of similarity measures among news, referring to textual contents, entity occurrences, and temporal ordering. These similarities are captured in a multi-view attributed graph. To distill canonicalized events, EVIN coarsens the graph by iterative merging based on a judiciously designed loss function. To infer semantic classes of events, EVIN uses statistical language models. EVIN provides a GUI that allows users to query the constructed knowledge base of events, and to explore it in a visual manner.

Categories and Subject Descriptors

H.1 [**Information Systems**]: Models and Principles

1. INTRODUCTION

Motivation: A named event is an entity that happened at a certain time point or during a certain time period. Examples include battles, elections, concerts, championships, disasters, fairs, summits, etc. Named events are particularly important for knowledge bases such as `dbpedia.org`, `yago-knowledge.org`, or `freebase.com` which is the core of the Google Knowledge Graph. These knowledge bases already contain entities of type event, fine-grained semantic classes to which they belong, and facts about them such as temporal positioning and location of events, involved people, etc. This kind of knowledge enables a new semantic awareness in search, summarization, analytics, and recommendations.

However, knowledge bases have fairly limited coverage of events, as they mostly rely on extracting entities and facts from Wikipedia and similar curated sources. None of the major knowledge bases taps into online news for increasing its population of named events. There is no methodology for harvesting events in the long tail, such as concerts of independent bands, and for capturing brand-new events, such as hurricanes or political scandals. For example, we would like to automatically capture an event like the *UEFA Champions League Final 2013*, along with its involved entities such as the winning team Bayern Munich, and also a subsequent event like the *Bayern Munich Triple*, referring to this team's winning of three championships a week later. The former is meanwhile captured in a Wikipedia article, but the latter is too specific for Wikipedia yet would be desirable to include in a high-coverage knowledge base.

Problem Statement: The EVIN (Events In News) project addresses the problem of populating a knowledge base with fine-grained events from news, along with detailed semantic typing (e.g., using classes like rock concerts or hurricanes), relationships among events (sub-events, temporal order, etc.), as well as people and organizations participating in events.

Prior Work: There are prior methods for mining events and storylines from a batch or stream of news articles (e.g., [2, 3, 7]). However, the problem with these methods is that their output contains high redundancy, e.g., by yielding 5 or 10 semantically overlapping events such as "Bayern wins Champions League", "Bayern beats Borussia", "Bayern's triumph in London", "Bayern overcomes last year's final trauma", etc. Instead, we would like to reconcile all these cues into a single named event in a canonicalized representation. Moreover, the prior works' output lacks semantic organization; none of the methods would attach fine-grained class labels such as *soccer finals* or *European sports championships* to the event. So there is a fundamental difference between traditional event mining and the goal of populating a knowledge base, as the latter requires semantic rigor and does not tolerate semantic redundancy.

Contribution: The EVIN project has developed methods and software tools for populating event classes (concerts, ceremonies, elections, conflicts, accidents, disasters, tournaments, etc.) of high-quality knowledge bases by extracting, cleaning, and canonicalizing fine-grained named events from news corpora. Our methodology involves the following components:

- mapping news articles onto semantic event classes;
- a multi-view graph model that captures relationships and relatedness measures between news items;
- a graph-coarsening algorithm for inferring canonicalized events and sub-events for knowledge base population;
- a GUI for querying, exploring, and visualizing the events in the knowledge base and their provenance.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577009.

2. NAMED EVENT EXTRACTION

News articles are modeled as sets of features that are used to compute different distance measures between news articles. Our task entails a *grouping* and a *chaining* problem: combining thematically highly related news into a single event, and ordering related events along the timeline. For example, for the news related to "UEFA Champions League 2012/2013" theme, ideal groups would be "Champions League draws", "Group matches", "Quarter-final matches", "Semi-final matches", and "Champions League final". Moreover, events should be chained among each other in temporal order. For instance, the event "Champions League draws" should be chained to the event "Champions League group matches". To jointly solve the problem of grouping and chaining, we have devised a novel graph coarsening technique applied to a multi-view attributed graph of news articles.

2.1 Computational Model

Feature sets: We exploit four kinds of feature groups that are provided by each news article N or can be derived from it by information extraction and other methods:

- the *textual content* of the article d,
- the *publication date* t,
- the *set of entities* $\mathcal{A}$ (people, organizations, etc.) appearing in the content, and
- the *semantic types* $\mathcal{T}$ of events covered by an article.

The entity names appearing in a news article are extracted using the Stanford NER tagger. As these are used as features for subsequent processing, we do not attempt to disambiguate entities onto canonical representations in a knowledge base, but simply accept a tolerable level of ambiguity and noise. Nevertheless, any named entity disambiguation method can be used to link entities onto canonical representations. We obtain ontological types for news articles by constructing statistical language models (LM's) for articles and types and mapping an article to similar types using Kullback-Leibler divergence. This is further discussed in Subsection 2.2.

Features are used to compute different kinds of similarity measures between news articles:

- The *content distance* is the cosine of the articles' $tf \cdot idf$ vectors over a bag-of-words model:

$$dist_{text} = cosine(\mathcal{V}(n_i), \mathcal{V}(n_j))$$

  where $\mathcal{V}(n_i)$ and $\mathcal{V}(n_j)$ are the $tf \cdot idf$ vectors of news articles n_i and n_j, respectively.

- The *temporal distance* is the normalized time distance between the publication dates of news articles:

$$dist_{temp}(n_i, n_j) = \frac{|t(n_i) - t(n_j)|}{H}$$

  where H is the time horizon of the corpus. H is calculated as the difference between the earliest and the latest timestamps in the corpus.

- The *entity distance* between articles is the weighted Jaccard coefficient capturing the overlap of the entity sets:

$$dist_{ent}(n_i, n_j) = \frac{\sum\limits_{a \in \mathcal{A}(n_i) \cap \mathcal{A}(n_j)} weight(a)}{\sum\limits_{a \in \mathcal{A}(n_i) \cup \mathcal{A}(n_j)} weight(a)}$$

  where the weight of an entity is its $tf \cdot idf$ value.

- The *type distance* between articles is the weighted Jaccard coefficient capturing the overlap of the type sets:

$$dist_{type}(n_i, n_j) = \frac{\sum\limits_{\tau \in \mathcal{T}(n_i) \cap \mathcal{T}(n_j)} weight(\tau)}{\sum\limits_{\tau \in \mathcal{T}(n_i) \cup \mathcal{T}(n_j)} weight(\tau)}$$

  where the types are weighted by their *idf* values.

Multi-view attributed graph (MVAG): All these components are used together to construct a multi-view attributed graph (MVAG) of news articles such that a vertex v is a news item with a set of attributes: its timestamp, its associated entities $\mathcal{A}$ and its types $\mathcal{T}$. The graph has two kinds of edges: undirected ones and directed ones. Two vertices are connected by undirected edges if they share at least one entity and at least one type. The weight of the edge is the content distance between two vertices. Two vertices are connected by a directed edge if their timestamps indicate that they are ordered on the timeline. The weight of a directed edge is the temporal distance between the vertices.

2.2 Mapping News to Semantic Classes

We have devised a two-step approach to map news articles to ontological event types. First, news are mapped to Wikipedia categories using statistical language models (LM's). Second, Wikipedia categories are mapped to WordNet event classes by the heuristic method of [8].

The first step is based on a linearly interpolated language model for news articles and categories. The LM of a news article is constructed as a mixture model of two LM's capturing i) the news contents in terms of title and body words, and ii) the entities including normalized date literals that appear in the article. The LM of a Wikipedia category is defined analogously and constructed from all Wikipedia articles that belong to the category.

The LMs are estimated in a standard way with Jelinek-Mercer smoothing (using the overall features in the entire collection). To compare how close a Wikipedia category is to a news article, we use the Kullback-Leibler (KL) divergence between the corresponding LM's. This is a standard measure in IR and NLP. For each news article, we compute the top-k categories based on this distance measure, and accept those categories whose KL divergence is above some threshold.

The second step of the two-step approach then maps the accepted Wikipedia categories to their lowest event type in the WordNet taxonomy DAG. The method of [8] parses the category name to identify its head word, and then maps this to the best matching WordNet type. If the head word is ambiguous, the sense frequency information of WordNet is used to make the choice. For example, for the category name "*General elections in Italy*", the word "*elections*" is found as the head word, and it is mapped to "*wordnet_election*".

2.3 From News to Named Events

The MVAG of news articles is coarsened in a way that the nodes reporting on the same event will be merged. The decision whether to merge two nodes is driven by a judiciously designed loss function. Given a multi-view attributed graph G with node set V, undirected edges E, directed edges F, entity sets $\mathcal{A}$, and types $\mathcal{T}$, where all components are weighted, EVIN computes a coarser graph G^* with $V^* \in 2^V$ (i.e., forming equivalence classes of nodes) and E^*, F^*, A^*, T^* such

Semantic Classes			
scandal	insurgency	show	final
crisis	attack	election	protest
occupation	festival	outbreak	conflict
riot	carnival	competition	tournament
bombing	inauguration	conference	show
flood	war	concert	recession
referendum	treaty	acquisition	tour

Table 1: 28 examples of semantic classes out of 217.

that i) G^* largely preserves the properties of G and ii) G^* is simpler (smaller) than G. The grouping of the original V nodes that leads to V^* induces the undirected edges E^* and directed edges F^* of the coarsened graph. This updates the edge weights as well as attribute sets. Figure 1 illustrates the coarsening. One can think of the coarsening as a form of clustering. However, the multi-view features of the graph, with two kinds of edges, makes this a non-standard problem.

Loss function: The loss function calculates the information lost when two or more nodes are merged into a coarser node. The function aggregates the loss over the different views of news articles. The intuition is that merged nodes should have similar neighbors, similar entities, similar types, and nearby timestamps. EVIN computes the information loss when merging a node x_i into coarser node x^* by summing up the following contributions:

- $L_E(x)$, the weighted number of additional undirected edges that x_i takes after being merged into x^* (i.e., incoming edges that x_i did not have but inherits from the other nodes in x^*);
- $L_F(x)$, the weighted number of additional directed edges that x_i takes as part of x^*;
- $L_\mathcal{A}(x)$, the weighted number of additional entities that x_i takes;
- $L_\mathcal{T}(x)$, the weighted number of additional types that x_i takes.

Thus, the loss function is $L(x) = L_E(x) + L_F(x) + L_\mathcal{A}(x) + L_\mathcal{T}(x)$.

Optimization model: The algorithms we developed to coarsen a MVAG aims to minimize an objective function that combines, by a linear combination, the *total information loss* and the *model complexity* of the MVAG, where the latter is measured in the size of the resulting graph. As the coarsening problem is likely NP-hard, we use greedy heuristics and stochastic optimization techniques.

A coarsened MVAG does not only contain coarse nodes of news articles, but also the chaining information between the events via directed edges, and the most important semantic types, entities of events via induced attribute sets. Please note that the technical details of the loss function, the objective function, and the optimization model are beyond the scope of this demo paper.

2.4 Populating the Knowledge Base

In order to populate a high-quality knowledge base, not all the events found by the coarsening algorithms are accepted. Only the events that have a gain of the objective function above a specified threshold are accepted. Each event is labeled by a "*representative news headline*". The representative news is chosen based on the centrality score in the original MVAG. For the purpose of this demo, we used 13,000 news articles, from a large variety of newspapers and online sources, crawled by following external references in the Wikipedia articles about 700 major events from 2010 to 2013. From this input, EVIN built a knowledge base with 1,072 named events, organized in 217 different semantic classes. Some examples of the semantic classes can be seen at Table 1.

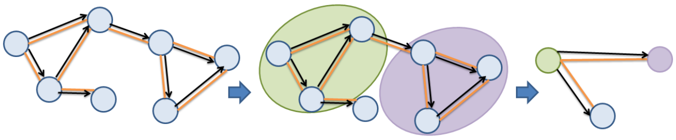


Figure 1: Coarsening a multi-view attributed graph.

3. DEMONSTRATION OF THE SYSTEM

3.1 Implementation

EVIN is implemented in Java and uses the Stanford NER tagger for detecting entity names.

EVIN computes the semantic types of news items in the corpus, all similarity measures between news items, builds and coarsens the MVAG for the news corpus. After the thresholding, the qualitative events are put into a knowledge base. Moreover, EVIN finds representative images for events via querying a search engine based on the important entities appearing in the event. The relevance of the images to the event shows the quality of the coarsening, as coarsening can successfully induce the important entities of an event. EVIN provides querying and explorative navigation by its Web-browser-based GUI over the knowledge base.

Figure 2 shows a screenshot of the EVIN prototype. The screenshot has three parts: the search and filter section at the top, the chaining band, and the grouping band. These serve the following purposes:

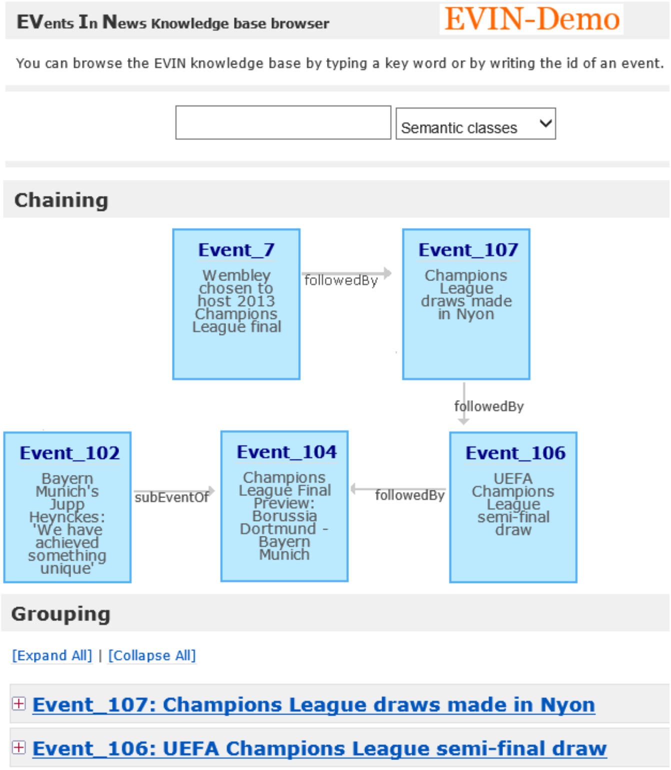
- EVIN uses keyword/semantic queries to retrieve events from the populated knowledge base.
- The chaining of events is shown at the chaining band.
- The grouping of events is shown in the grouping band.
- Upon clicking on an event, the lower band shows the semantic types, the entities, the images, and the news belonging to an event. The news items along with detailed info is also shown for the clicked event.

The EVIN system can be examined online at `http://www.mpi-inf.mpg.de/yago-naga/evin`. This demo uses a sample of about 100 events from the EVIN knowledge base, since these events are enriched by images via image search.

3.2 Scenarios

The system supports three ways of interactive search and exploration: keyword-based querying, navigation by semantic classes, and the combination of these two modes.

Keyword-based querying: Users type keyword queries such as "Champions League" to retrieve events matching the query. EVIN finds events like "*Event_7: Wembly chosen to host 2013 Champions League Final*", "*Event_102: Bayern Munich's victory*", "*Event_104: Champions League Final preview*", etc. The first event includes news articles about the UEFA announcing the venue of the final, the second event includes Bayern Munich's victory, the third event includes news from the week before the final. Figure 2 shows the chaining and sub-event information between these



EVents In News Knowledge base browser

EVIN-Demo

You can browse the EVIN knowledge base by typing a key word or by writing the id of an event.

Chaining

Event_7
Wembley chosen to host 2013 Champions League final

followedBy

Event_107
Champions League draws made in Nyon

followedBy

Event_102
Bayern Munich's Jupp Heynckes: 'We have achieved something unique'

subEventOf

Event_104
Champions League Final Preview: Borussia Dortmund - Bayern Munich

followedBy

Event_106
UEFA Champions League semi-final draw

Grouping

[Expand All] | [Collapse All]

⊞ Event_107: Champions League draws made in Nyon

⊞ Event_106: UEFA Champions League semi-final draw

Figure 2: Screenshot of EVIN.

events. Figure 3 shows the grouping of the events and also news headlines as sub-items of the events. Detailed information can be viewed by clicking on items.

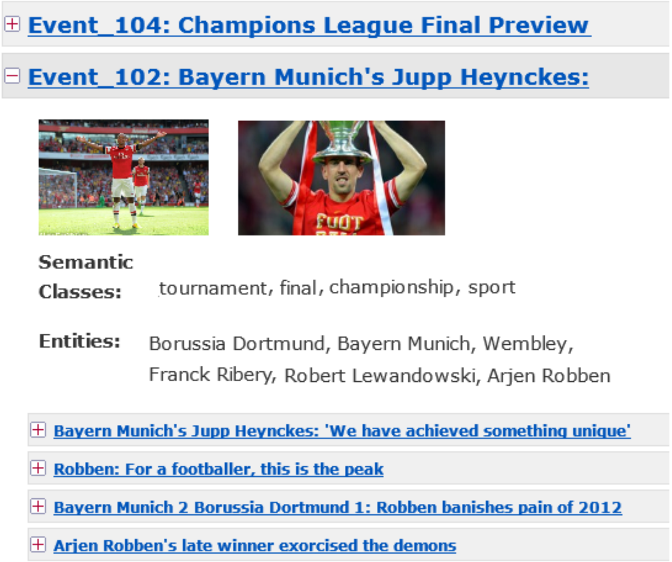
Navigating semantic classes: EVIN uses WordNet event classes as semantic labels. Currently, around 200 semantic classes are populated by the demo corpus. Examples are "*championship, contest, game, political campaign, tournament, final, riot, concert, conflict, war, election, invasion*", etc. The user can choose a semantic class from a drop-box menu in order to explore the events belonging to the class. For example, the semantic class "final" contains all the events about cup finals.

Combined search and exploration: By using the keyword-query box and the semantic-classes menu together, users can find events within a certain class. For example, the Champions League Final between Bayern Munich and Borussia Dortmund in 2013 can be easily retrieved by combining the query "German rivals" and the semantic class "final". There are only two events in the result set: "*Event_102: Bayern Munich's victory*" and "*Event_104: Champions League final preview*".

4. RELATED WORK

Story mining from news: There is large amount of work on topic detection and story mining on news, e.g., [3, 7, 9, 6]. However, the events found by these systems are not ontological events and not clean enough for populating a knowledge base. Knowledge bases i) require canonical representations and ii) semantic typing of the events, which are among the objectives of the EVIN project.

Graphs for events: Identifying events in news streams has been addressed by modeling news articles as graphs of entities [2] or as graphs of keywords [1]. Both methods identify dense clusters in graphs. However, an event here is merely a set of temporally co-occurring entities and/or keywords. Thus, events are only implicitly represented; they are not canonicalized and cannot be used to populate a knowledge base.



⊞ Event_104: Champions League Final Preview

⊟ Event_102: Bayern Munich's Jupp Heynckes:

Semantic Classes: tournament, final, championship, sport

Entities: Borussia Dortmund, Bayern Munich, Wembley, Franck Ribery, Robert Lewandowski, Arjen Robben

⊞ Bayern Munich's Jupp Heynckes: 'We have achieved something unique'

⊞ Robben: For a footballer, this is the peak

⊞ Bayern Munich 2 Borussia Dortmund 1: Robben banishes pain of 2012

⊞ Arjen Robben's late winner exorcised the demons

Figure 3: Grouping of the news into events.

Mapping documents to Wikipedia categories: There are numerous studies on using Wikipedia categories for clustering documents, e.g., [4, 5]. These methods exploit the semantic relatedness of documents to Wikipedia concepts and the link structure of Wikipedia categories. Our approach uses statistical language models to map news articles to YAGO classes, via specific Wikipedia categories.

5. REFERENCES

[1] M. K. Agarwal, et al. Real Time Discovery of Dense Clusters in Highly Dynamic Graphs: Identifying Real World Events in Highly Dynamic Environments. *PVLDB*, 2012.

[2] A. Angel, et al. Dense Subgraph Maintenance under Streaming Edge Weight Updates for Real-time Story Identification. *PVLDB*, 2012.

[3] A. Das Sarma, et al. Dynamic Relationship and Event Discovery. WSDM, 2011.

[4] E. Gabrilovich, et al. Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *AAAI*, 2006.

[5] X. Hu, et al. Exploiting Wikipedia as External Knowledge for Document Clustering. *KDD*, 2009.

[6] Y. Rui, et al. Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution. In *SIGIR*, 2011.

[7] D. Shahaf, et al. Connecting the Dots Between News Articles. In *KDD*, 2010.

[8] F. M. Suchanek, et al. Yago: A Core of Semantic Knowledge. WWW, 2007.

[9] D. Wang, et al. Generating Pictorial Storylines Via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *AAAI*, 2012.