# Using Uncertain Graphs to Automatically Generate Event Flows from News Stories

Laura Christiansen, Bamshad Mobasher, and Robin Burke
Center for Web Intelligence
DePaul University
lchris10@cdm.depaul.edu, mobasher@cs.depaul.edu, rburke@cs.depaul.edu

## Abstract

*Capturing the branching flow of events described in text aids a host of tasks, from summarization to narrative generation to classification and prediction of events at points along the flow. In this paper, we present a framework for the automatic generation of an uncertain, temporally directed event graph from online sources such as news stories or social media posts. The vertices are generated using Natural Language Processing techniques on the source documents and the probabilities associated with edges, indicating the degree of certainty those connections exist, are derived based on shared entities among events. Graph edges are directed based on temporal information on events. Furthermore, we apply uncertain graph clustering in order to reduce noise and focus on higher-level event flows. Preliminary results indicate the uncertain event graph produces a coherent navigation through events described in a corpus.*

## 1 Introduction

Extracting a narrative progression from text opens the door for a host of useful applications. Representations of the key stories can be simplified or expanded upon to aid comprehension. Examining the dynamics of the narrative events can reveal emergent information and points of change that may be useful not only in understanding the story but in predicting future dynamics. One can observe how paths differ when looking at different domains, such as news sources versus social media, providing insight in how both represent events. Understanding the flow of information over time is valuable.

Intuitively, we understand that flow is not flat. One event may branch out to connect to events later in time. Likewise, many events may feed into a single event. An extracted timeline for a narrative will capture the temporal ordering but lose information on the connections between events. On the other hand, evidence of a connection between events may be incorrect. Inferring connections can lead to differing levels of certainty in the likelihood of those connections. This lends itself to probabilistic or uncertain graphs, where edges have probabilities of their existence. An uncertain graph is not discrete but is rather a template to generate all ”possible worlds”: all discrete graphs that are drawn from the edge probabilities.

We propose a framework for automatically extracting an uncertain event graph, with edges directed by the temporal flow of events, from online sources such as news stories or social media posts. The first stage of this process involves event extraction using natural language processing (NLP) techniques. Part-of-speech (POS) tagging and semantic role labeling (SRL) allow us to extract predicates, which we treat as events. Entity detection labels what named entities are involved in those events while temporal expression extraction defines the temporal ordering between events. Next, we generate the edges and their probabilities based on Bayesian combination of evidence. As basic evidence, we use the extracted entities shared between events and the proximity of event references within the text. We focus on simple, text-based evidence of events and their connections but more complex information derived from metadata can be utilized. Different domains offer different possible sensors for detecting and tracking events. This process generates the vertices and edges of the uncertain graph, and those edges can then be directed based on the temporal ordering information discovered in the first stage.

Once the full event graph is generated, we use uncertain graph clustering to reduce noise and discover higher-level abstractions, with clusters indicating closely related events describing a larger, meta-topic within the graph. We use pKwikCluster, a clustering algorithm for uncertain graphs, to identify likely clusters. As a precursor to a larger user study evaluation, we observe the flow and connections within the graph to evaluate its coherency and correctness. Our preliminary results on a dataset consisting of news articles indicates this is a viable approach to automatically capturing and depicting a branching flow of events.

## 2 Related work

Linking and tracking events is a research problem that has been addressed from a number of angles. Extending their previous work in event extraction, Rospocher et al. [1] propose a approach for automatically generating knowledge graphs based on the discovered events. In this knowledge graph, edges are predicates and nodes are entities as opposed to combining both within an event construct. Moving in the opposite direction, Althoff et al. [2] generate timelines from knowledge graphs. The generated timelines are personalized and provide a temporal ordering but not branching connections. Shahaf et al. [3] developed an algorithm for generating zoomable, intersecting timelines of key terms to summarize news. These timelines are constructed in relation to each other and the terms that make up the nodes are annotated with news stories and is only intended for a high-level event representation.

Event detection and extraction has been approached in a number of ways. Work in [4] discovers a specific type of event, earthquake occurrences, from microblogs. User chatter on earthquakes is classified and filtered to act as sensors to determine when and where an earthquake strikes. Also working with microblogs, [5] clusters tweets based on keywords and locations to detect new events. Keywords, combined with the time and place they were posted, form a rough event reference. In [6], events are automatically pulled from streaming news data; news relations are extracted then clustered to find different representations of the same event before training a model to extract news relations based on that co-reference information. This attempts to overcome issues of different linguistic descriptions of the same event. We only address this issue indirectly, through co-reference resolution, but similarly are interested in basic relations in the form of verb predicates.

When constructing networks, there may be doubts regarding the accuracy of connections between nodes are due to the techniques used to construct those connections. Link prediction may be erroneous or sensors may have detected noise. Uncertain graphs tackle this problem by assigning probabilities of existence to edges; an uncertain graph is the template with which to generate a set of possible discrete graphs based on those probabilities. For example, Zhao et al. [7] use uncertain graphs to detect protein complex structures. In their graphs edges are interactions between structures, but there is noise in data related to when they interact. Prachas et al. [8], propose a method to generate the best discrete approximation from an uncertain graph. We avoided generating discrete graphs from our uncertain graph, relying on algorithms that approximate calculations over a discrete graph set. Clustering algorithms are extended to uncertain graphs by Kollios et al. [9], and we use their adaption of pKwikCluster and definition of estimated edit distance over an uncertain graph to aggregate our event vertices. Bonchi et al. [10] examine how to perform core decomposition in an uncertain graph context, an approach we did not use but may be useful in creating higher-level representations of an uncertain event graph.

## 3 Uncertain event graphs

In this section, we describe our method for constructing a temporally directed, uncertain event graph. First, we extract the necessary information from text using a variety of NLP techniques to construct the event vertices. Once events are defined, we proceed to the definition of edges, their probabilities, and their direction. Finally, we aggregate events within clusters to provide a higher-level representation of the event graph structure.

### 3.1 Event extraction

The first stage is to discover the events described in the data. At a basic level, this process includes the identification of an action that occurs and the entities involved. To this end, we define our initial event references in terms of predicates. Predicates define actions within a sentence and serve as an anchor point for additional details involving the subjects and objects. To extract this information from text, we need to run POS tagging and dependency parsing. This identifies which parts of speech the different terms in a sentence have as well as identifying the sentence structure and with SRL, the roles entities play within a predicate can be further identified. Take the sentence ”John bought a car in Boston”; using dependency parsing and SRL, we can identify ”bought” as the predicate verb, ”John” as the subject, ”a car” as the object and ”in Boston” is the location. We consider predicates references to events rather than events; this distinction is important as the same event may have multiple references.

Co-reference resolution, another established NLP task, enables discovery of multiple representations of the same event within the same document. Two predicates co-referencing each other indicate the same event is discussed. Our definition of an event can now encompass multiple predicates based on co-references. To expand our example, if another sentence read ”He bought it last Friday”, co-reference resolution can tell us if this instance of ”bought” is referring to the same event as the predicate verb in the first event. Similarly, it can tell us if ”he” refers to ”John”. This helps better define the entities involved in an event; in our event reference we can substitute the more informative proper noun for the pronoun.

This substitution is further enhanced by Named Entity Recognition (NER). NER identifies and classifies of named entities as people, locations, or organizations. To continue our example, NER would identify ”John” as a person and ”Boston” as a location. Combining this with co-reference resolution, we can find all co-references to a named entity and include the entity information in those events.

Finally, the temporal relationships between events can be ascertained through temporal expression extraction. In some cases, this finds the fixed time interval described in the text. In others, it is is relative. The exact date of event $E_1$ might not be known but we know it took place before event $E_2$. be places parts of the text in time. Another sentence might tell us "Afterwards, John bought coffee"; we can label the coffee purchase event as occurring after the car buying. By knowing this, we know the temporal flow of events.

We use the English language version of the Newsreader [1] pipeline to perform these NLP tasks on our dataset. Described in [11], the pipeline is a series of NLP modules intended primarily for news text. We are not using the output of the entire pipeline; instead, our focus is POS tagging, dependency parsing and SRL, co-reference resolution, NER, and temporal relations. Each event has at least one predicate representation and includes information on the roles within that predicate as well as any named entities involved. If an event contains no entities, it is removed. This describes our event extraction from within a single text source.

$$J(A,B) = \frac{(A \cap B)}{(A \cup B)} \tag{1}$$

The same events may also be referenced between documents, which is not identified by the techniques described. To tackle this, we first look for the date range of the events. Events whose known time intervals overlap are candidates to be combined. We also include events without an explicit time interval but whose document publication dates are within a day of each other. This extension makes sense in the context of news articles but should be omitted or replaced for other datasets. Candidates for merging then have their term sets compared via Jaccard similarity, defined in equation 1. These term sets are pruned to exclude conjunctions, articles, and punctuation. Any candidates with a Jaccard similarity greater than threshold $\alpha$ are combined.

### 3.2 Edge generation

The vertices in the uncertain graph are the events we've just described. The next stage is to generate the edges in the graph graph. For this preliminary work, we examine basic relationships indicating two events are connected. As we are constructing an uncertain graph, this requires computing the probability that a link between two events exists given the evidence at hand. Entity similarity and document co-occurrence proximity between events are the types of evidence we use in the proposed approach. The first measure examines whether the same entities are involved in two events. We posit two events sharing entities are more likely to be related than those that don't. The second measure, intra-document proximity, operates on the assumption that an author is not jumping from tangent to tangent within their writing; the closer two the descriptions of two events are within the text of a document, the more related we can assume those events are.

Given those assumptions, we define the probability of a link existing between two events given their entities and intra-document proximity. Assuming both sources of evidence are conditionally independent, we calculate the probability of a link existing given their evidence with equation 2 using Bayes rule. Let $L$ represent whether two events are linked, $En$ the shared entities between events, and $D$ the the intra-document proximity between events. For $P(L)$, we assume an ignorant prior of 0.5.

[1] http://www.newsreader-project.eu/

$$
\begin{aligned}
P(L|En,D) &= \frac{P(L)P(En,D|L)}{P(En,D)} \\
&= \frac{P(L)P(En|L)P(D|L)}{P(En,D)}
\end{aligned}
\tag{2}
$$

where

$$P(En,D) = P(L)P(En|L)P(D|L) + P(\neg L)P(En|\neg L)P(D|\neg L)$$

$P(En|L)$ is defined here as the average Jaccard similarity between the predicates of two events.This is an average as, either through co-referencing predicates or combination of events between documents, an event can have more than one predicate representation. In equation 3, $m$ and $n$ represent the number of predicates describing events $E_1$ and $E_2$ respectively, while $En_i$ and $En_j$ refer to the entity set for predicates $p_i$ and $p_j$. $J(En_i, En_j)$ is the Jaccard similarity between entity sets $En_i$ and $En_j$.

$$P(En|L) = \frac{1}{(m \times n)} \left( \sum_{i=1}^{m} \sum_{j=1}^{n} J(En_i, En_j) \right) \tag{3}$$

$P(D|L)$, referring to the intra-document proximity, is defined in equation 4. Here, $dist(p_i, p_j)$ is a simple measure of the distance between the sentences in which predicates $p_i$ and $p_j$ occur, while $t_d$ is the total number of sentences in document $d$.

$$P(D|L) = \frac{1}{(m \times n)} \left( \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{dist(p_i, p_j)}{t_d} \right) \tag{4}$$

Additive smoothing of 0.1 is applied to equations 3 and 4 so that event pairs with a probability of 0 in one are not immediately removed from the graph.For each pair of events, the value of $P(L|En, D)$ is calculated. In the uncertain graph, this represents the probability the two events are linked and subsequently that an edge exists. Additional evidence can easily be incorporated using Bayes' rule, extending equation 2. Taking into account the temporal information extracted during the NLP stage, we can direct the uncertain graph edges based on the temporal information. This is either done by comparing the explicit time interval or through relative temporal relations. Edges are omitted from the graph if we have no temporal information and the edge is directed from the earlier to the later event. If both events occur simultaneously, then the edge is bidirectional. Events without any possible edges are pruned from the graph. We now have a temporally directed uncertain graph of events.

### 3.3 Event abstraction

For ease of quickly interpreting a large event graph, some degree of aggregation and abstraction is useful. It provides a simpler representation and further information on how the events are related to one another. To accomplish this aggregation, we turn to graph-based clustering. This is complicated by as the event graph is not discrete but rather an uncertain graph that can be used to generate a large set of discrete graphs.

In [9], clustering methods were extended to apply to uncertain graphs. We borrow their pKwikCluster, an adaptation of kwikCluster, to find event clusters within the graph. The pKwikCluster algorithm is simple: pick an available vertex at random as a new cluster, add to that cluster all available neighbors with an edge probability greater than 0.5, then mark all vertices in the new cluster as unavailable. Repeat these steps until every vertex is part of a cluster. This algorithm is run multiple times and the results of each run are compared to determine which produced the best clustering.

The goodness of a clustering result can be evaluated, in part, by the edit distance. For the non-probabilistic kwikCluster, this is between the graph and the cluster graph. Assume all edges between clusters are omitted; the edit distance indicates how many changes needed to be made to the base graph's structure to accommodate that result. The clustering run that minimizes this metric is selected.

$$D(\mathcal{G}, \mathcal{Q}) = \sum_{\{u,v\} \in E_{\mathcal{Q}}} (1 - P_{uv}) + \sum_{\{u,v\} \notin E_{\mathcal{Q}}} (P_{uv}) \tag{5}$$

In an uncertain graph context, this becomes the edit distances between the cluster graph and all possible worlds generated by the uncertain graph. Rather than compute that daunting metric, we can instead compute a single estimated edit distance between the uncertain graph and cluster graph. This is shown in equation 5. $P_{uv}$ is the edge probability between vertices $u$ and $v$, $E_Q$ refers to all edges within a cluster, $\mathcal{Q}$ is the set of clusters, and $\mathcal{G}$ is the uncertain graph.

## 4 Results

We ran our framework on a selection of 190 news articles from The Guardian [2] covering the US election. These articles covered a range of dates from January 2015 to January 2017. Clustering was run 100 times and the run with the smallest edit distance for its cluster assignments is the clustering described here. As discussed earlier, events are pruned from the graph if they lack entity information and edges are pruned if they lack information on their temporal direction. We omitted edges with probabilities lower than 0.1 and event vertices with no connecting edges. This results in 1606 events and 15779 edges. Figure 1 shows a visualization of the uncertain graph. The nodes and edges are shaded based on cluster assignments for events; there are 1053 clusters in total.

In Figure 2, we've labeled some of the main clusters within the graph based their focus on the candidates. Table 1 describes the major events from the clusters pertaining to candidates Marco Rubio and Chris Christie; if an event is listed as having an edge with another event ID, that indicates the event points to that event. This event either preceded the other unless a complementary edge exists in the opposite direction; in that case, the events occurred simultaneously.

We can follow the branching path through these events. Events 1-8 apply to Rubio while 8-10 are focused on Christie and we can observe how the flow of this subgraph moves, as well as how these two clusters connect. Figure 3 visualizes this subgraph. For example, Rubio discussed childhood taunts directed at family in event 3, which occurred at roughly the same

[2] http://www.theguardian.com

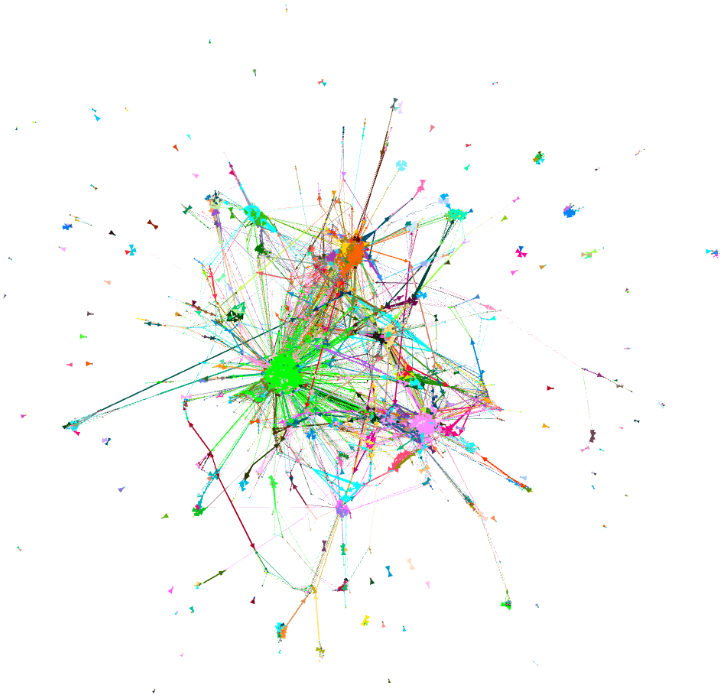

Figure 1: Uncertain graph for US election dataset.

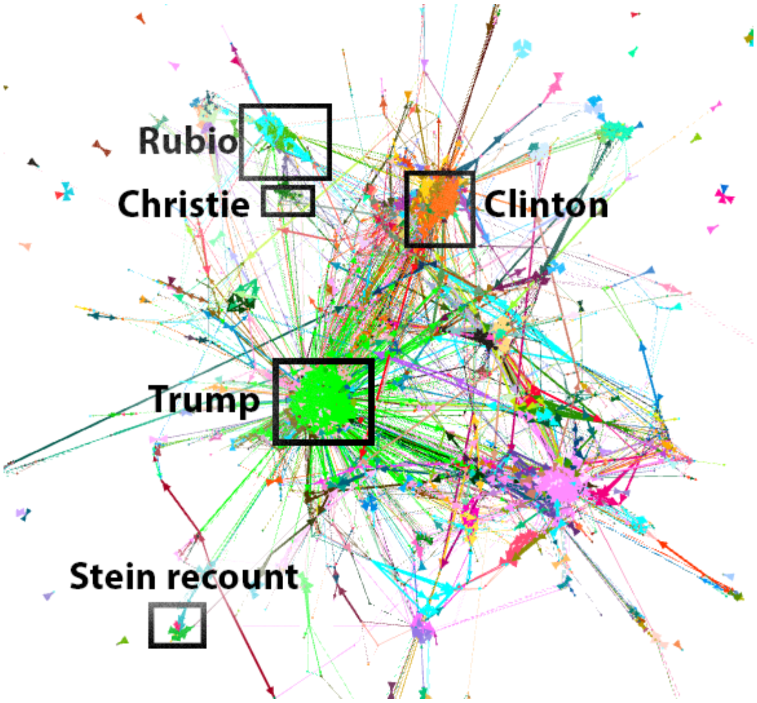

Figure 2: Partially labeled uncertain graph

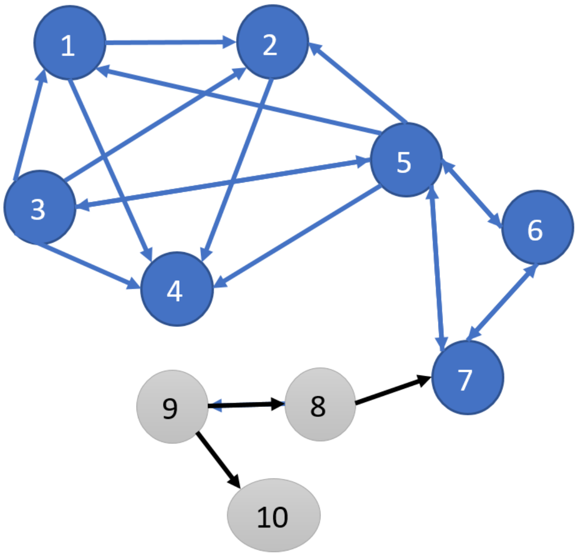

Figure 3: Subgraph for Rubio and Cruz.

time as he was characterized as the mainstream Republican candidate with the best chance of winning the nomination in event 5. Event 3 also took place before the discussion of attack ads against Rubio in event 1, the attack Donald Trump for having small hands in 4, and the suggestion Rubio should bow out of the primaries in event 2. The Rubio and Christie clusters meet at the point where the discussion shifts to what the candidates will do given the New Hampshire primaries. Further, we can see that event 8 is speculating before that primary while event 7, which came after, is discussing the results. The least likely edge in the subgraph was between 8 and 9, with a probability of 0.20861. These are events that are both discussing Christie but only one explicitly includes Christie as an entity; similarly, they co-occur in the same document but not particularly near each other.

Turning our attention to a clustered area further away from the center of the central component, we can see how events related to the recount Jill Stein funded in three states appear in the graph. Table 2 lists the events and their edges while figure 4 illustrates the connections. This subgraph contains events from two separate clusters as events 2-4 are in one and event 1 is in another. Event 1, a recount being initiated in Wisconsin, is captured as co-occurring in time with event 2, in which Jill Stein requesting recounts in multiple states. Event 2 branches into 3 and 4. Event 3 covers the funds Stein raised throughout her campaign to trigger recounts, which the graph shows as occurring after Stein's initial recount requests and event 4, Stein discussing the Trump campaign's effort to stop the recounts. While events 1 and 2 share terms, they share no entities, so that edge is based solely on intra-document proximity.

The scattered outer circle of events and edges in figure 1 is comprised of smaller connected components and provides a useful illustration of the weaknesses of the limited pool of evidence we currently consider when constructing the graph. Often, these components are comprised of events that lack neighbors because of insufficient temporal information and mismatching entities. For example, the events described by ”fellow candidate Ben Carson is leaving the race” and ”caucus (or vote) for Cruz” are connected with a probability of 0.27675; they

| ID | Event | Edges |
|---|---|---|
| 1 | ads attacking Rubio | 2,4 |
| 2 | Rubio should bow out | 4 |
| 3 | [Rubio's] mom doesn't even swim | 1,2,4,5 |
| 4 | Rubio has accused him of having small hands | - |
| 5 | when [Rubio] is characterized [...] | 1,2,3,4,6,7 |
| 6 | similar candidates have had to drop | 5,7 |
| 7 | [Rubio] comes in the top three in New Hampshire | 5,6 |
| 8 | a swift exit after New Hampshire seems likely | 7,9 |
| 9 | money talks so don't count [Christie] out yet | 8,10 |
| 10 | another party heavyweight once tipped to go far | - |

Table 1: Events from Rubio and Christie clusters

| ID | Event | Edges |
|---|---|---|
| 1 | a recount has been initiated in Wisconsin | 2 |
| 2 | [Stein] requesting the recounts | 1,3,4 |
| 3 | the large funds Stein has raised throughout this process | - |
| 4 | the Trump campaign's cynical efforts to delay the recount [...] are shameful and outrageous," Stein said | 3 |

Table 2: Subgraph for Stein recount

share no entities but co-occurred in a document. What they describe is an event during the Iowa caucuses where candidate Ted Cruz told voters Ben Carson had left the race, so Carson voters should vote for Cruz. The connection is valid and the direction, from the former to the latter, is accurate. If anything, it makes intuitive sense that the link should be stronger but the entity mismatch was detrimental. These smaller components also capture tangents to the larger theme of the dataset. One pair of events, "she was arrested in Cairo" and "Egyptian courts would let her go free", refers to a US citizen, Aya Hijazi, who was arrested in Cairo. One of the articles in the dataset consisted of excerpts for a number of current stories, some of which were not related to the election. The graph generation correctly identified these two

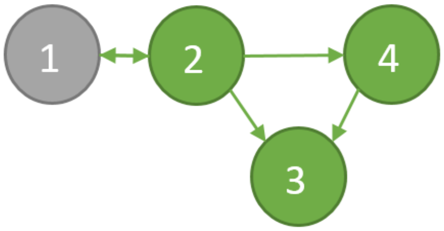

Figure 4: Subgraph for Stein recount.

events were related but, understandably, could not connect them to the main component.

Our initial uncertain graph of events is coherent and appears to provide a good temporal flow through the graph; we have maintained that flow while allowing event paths to branch. The clustering as a form of aggregating events was useful in analyzing the graph. Any issues with the clustering would seem to be an issue with the factors involved in linking events, perhaps overrating some edges between nodes with only a single named entity or underrating edges that lack entity overlap. Some of the weaknesses described would appear to be solvable by the inclusion of more text to build the graph, which would provide further definition connections for the event references. Another possibility would be to expand the definition of an event reference; we have the dependencies from the text and can represent more structure than individual predicates. Finally, we began with two forms of evidence but more might be useful. Incorporating additional sources could strengthen connections between nodes that lack significant entity overlap. The importance of this is likely increased when not dealing as noise in the dataset increases.

## 5 Conclusion and future work

We have presented a novel approach to automatically generate an uncertain event graph from a text dataset and shown anecdotally how the results are cogent and accurate. The dataset used here consists of online news articles but the proposed approach could be applied other online sources such as social media posts from which events and entities can be extracted. Our immediate next steps are to augment our initial definition of an event to pull in more sentence structure and incorporate additional forms of evidence in addition to named entities and intra-document proximity. From there, we will more comprehensively evaluate the accuracy of the uncertain graph via a user study, asking participants to assess whether pairs of events are connected and comparing their aggregate results with the uncertain graph probabilities. Finally, we intend to use the graph to predict dynamics and links within the graph and to examine how to create comparable graphs from social media sources.

## References

[1] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, T. Bogaard, Building event-centric knowledge graphs from news, Web Semantics: Science, Services and Agents on the World Wide Web 37 (2016) 132–151.

[2] T. Althoff, X. L. Dong, K. Murphy, S. Alai, V. Dang, W. Zhang, Timemachine: Timeline generation for knowledge-base entities, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 19–28.

[3] D. Shahaf, J. Yang, C. Suen, J. Jacobs, H. Wang, J. Leskovec, Information cartography: creating zoomable, large-scale maps of information, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 1097–1105.

[4] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 851–860.

[5] H. Abdelhaq, C. Sengstock, M. Gertz, Eventweet: Online localized event detection from twitter, Proceedings of the VLDB Endowment 6 (12) (2013) 1326–1329.

[6] C. Zhang, S. Soderland, D. S. Weld, Exploiting parallel news streams for unsupervised event extraction, Transactions of the Association for Computational Linguistics 3 (2015) 117–129.

[7] B. Zhao, J. Wang, M. Li, F.-X. Wu, Y. Pan, Detecting protein complexes based on uncertain graph model, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 11 (3) (2014) 486–497.

[8] P. Parchas, F. Gullo, D. Papadias, F. Bonchi, The pursuit of a good possible world: extracting representative instances of uncertain graphs, in: Proceedings of the 2014 ACM SIGMOD international conference on management of data, ACM, 2014, pp. 967–978.

[9] G. Kollios, M. Potamias, E. Terzi, Clustering large probabilistic graphs, IEEE Transactions on Knowledge and Data Engineering 25 (2) (2013) 325–336.

[10] F. Bonchi, F. Gullo, A. Kaltenbrunner, Y. Volkovich, Core decomposition of uncertain graphs, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 1316–1325.

[11] P. Vossen, G. Rigau, L. Serafini, P. Stouten, F. Irving, W. R. van Hage, Newsreader: recording history from daily news streams., in: LREC, 2014, pp. 2000–2007.