StoryPivot: Comparing and Contrasting Story Evolution

Anja Gruenheid[†] Theodoros Rekatsinas[*] Donald Kossmann[†] Divesh Srivastava[+]

[†]*Systems Group*
Department of Computer Science
ETH Zurich
{agruen,donaldk}@inf.ethz.ch

[*]*University of Maryland*
thodrek@cs.umd.edu

[+]*AT&T Labs-Research*
divesh@research.att.com

ABSTRACT

As the world evolves around us, so does the digital coverage of it. Events of diverse types, associated with different actors and various locations, are continuously captured by multiple information sources such as news articles, blogs, social media etc. day by day. In the digital world, these events are represented through *information snippets* that contain information on the involved entities, a description of the event, when the event occurred, etc. In our work, we observe that events (and their corresponding digital representations) are often inter-connected, i.e., they form *stories* which represent evolving relationships between events over time. Take as an example the plane crash in Ukraine in July 2014 which involved multiple entities such as "Ukraine", "Malaysia", and "Russia" and multiple events ranging from the actual crash to the incident investigation and the presentation of the investigator's findings. In this demonstration we present StoryPivot, a framework that helps its users to detect evolving stories in event datasets over time. To resolve stories, we differentiate between *story identification*, the problem of connecting events over time within a source, and *story alignment*, the problem of integrating stories across sources. The goal of this demonstration is to present an interactive exploration of both these problems and how events can be dynamically interpreted and put into context in real-world datasets.

1. INTRODUCTION

Lately there has been an increasing interest in monitoring news media, blogs, and social media from all over the world and extracting geo-referenced records that correspond to different real-world events and interactions between diverse groups of people, international organizations, countries etc. The extracted event records are commonly stored in repositories that get updated regularly whenever new extractions are obtained from a diverse collection of data sources. Example repositories of such extractions include GDELT [11] and EventRegistry [9] which are updated over fixed time intervals (e.g., daily). Extracted data is stored in a tuple format containing for example information about its origin, the type of the corresponding real-world event describing an activity linked with the event, the entities associated with the corresponding activity, a short description and a timestamp. An example of such a tuple is <New York Times, Accident, {Ukraine, Malaysian Airlines}, "Plane Crash", 07/17/2014>.

Extracted pieces of information over a fixed time-window represent only a snapshot of what is happening in the world. Real-world events are often connected over time forming *stories* that represent evolving relationships between different entities. Recovering the evolution and the dynamics of news stories across time is of tremendous value in different application domains, ranging from trend detection to economic analysis, and has been a focus of recent research literature [2, 10]. Nevertheless, the majority of proposed approaches for story detection focus on identifying current and thus often mentioned stories in streaming news. Thus, they do not provide means for tracking the evolution and dynamics of stories even though this type of story tracking has applications in different domains. For example analysts rely on temporal patterns of event occurrences to discover supporting evidence and validate their hypotheses. The most common paradigm to consider is political scientists who rely on historical data to forecast political crises [18, 8] and conflicts [20]. A different application domain that is increasingly popular is that of financial analytics based on political event extractions [7, 6].

Another limitation of currently employed systems is that they are oblivious to the presence of multiple data sources and their story reporting *perspectives*. Data sources have different perspectives on stories because they report the same story with varying content and with varying levels of timeliness [21]. Further, it has been recently shown that leveraging these individual source characteristics can lead to a significant accuracy improvement for difficult prediction tasks such as predicting civil unrest incidents [14]. Moreover, there has been an increasing interest in the database community to understand and exploit the characteristics of different data sources to either provide affordable analytics or to improve the quality of integrated data [4, 19, 15].

In this demonstration, we present StoryPivot, a novel story detection framework for detecting stories involving different entities over time. StoryPivot allows users to explore the evolution and dynamics of stories but also to understand the perspectives of different data sources when reporting these stories. To address the limitations mentioned above, we decompose the story detection problem into two subproblems (1) *story identification* and (2) *story alignment*. Story identification is the problem of finding those events that represent the same real-world story within a single source, and captures the individual story perspectives of each data source. Story alignment then integrates stories across multiple sources to enhance completeness and correctness. Due to the sheer number of available sources, one of the main challenges here is combining stories across

SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.
Copyright © 2015 ACM 978-1-4503-2758-9/15/05 ...$15.00.
http://dx.doi.org/10.1145/2723372.2735356.

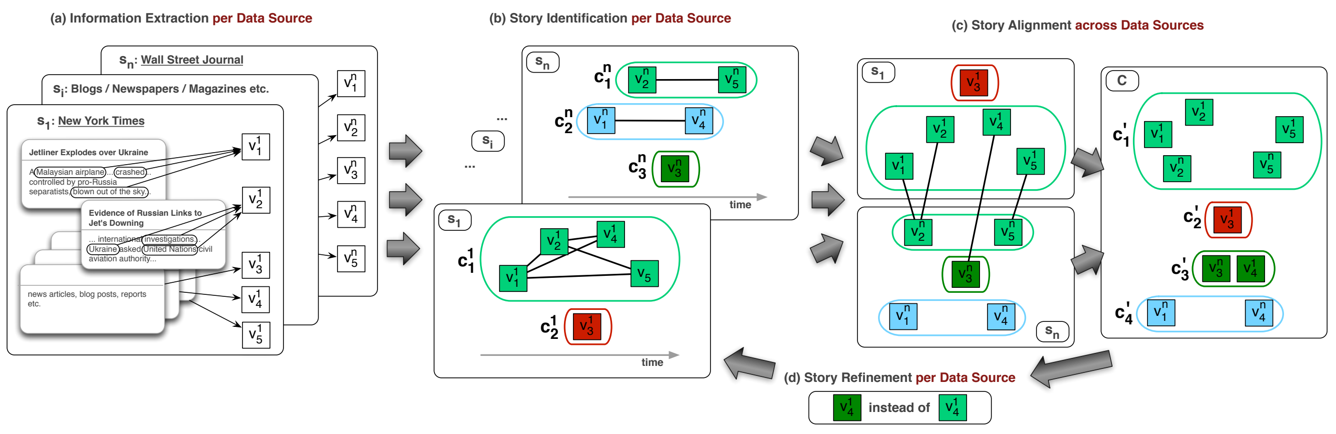

(a) Information Extraction per Data Source
(b) Story Identification per Data Source
(c) Story Alignment across Data Sources
(d) Story Refinement per Data Source
s_n: Wall Street Journal
s_i: Blogs / Newspapers / Magazines etc.
s_1: New York Times
Jetliner Explodes over Ukraine
A Malaysian airplane crashed controlled by pro-Russia separatists blown out of the sky.
Evidence of Russian Links to Jet's Downing
... international investigations Ukraine asked United Nations civil aviation authority...
news articles, blog posts, reports etc.
time
instead of

Figure 1: STORYPIVOT system overview.

data sources efficiently. In summary, we present the following features in our demonstration:

Interactive Framework. We provide an interactive interface for the users that allows them to examine the development of stories over time. It will provide visualization for both, story identification and story alignment, components of STORYPIVOT.

Large-Scale Experiments. To provide evidence of the scalability and efficiency of STORYPIVOT, we perform extensive experiments whose results are presented during the demo. Furthermore, the user can use the interactive visualizations to explore the results of these experiments as well.

The remainder of this demonstration description is structured as follows: First, we introduce the problem of story detection and a 2-phase mechanism to solve it in Section 2. It is applicable for a variety of use cases that we briefly describe in Section 3. Afterwards, we present the details on the technical demonstration in Section 4 and conclude this demonstration paper in Section 5.

2. STORY EVOLUTION

Identifying the evolution of stories requires detecting related events over multiple sources over time and combining them into a single cohesive story. In this section, we present an overview of STORYPIVOT which detects the evolution and dynamics of stories through processing event data. Furthermore, we present techniques on how to address story identification and story alignment efficiently.

2.1 Overview

An overview of STORYPIVOT is illustrated in Figure 1. Within the system, we use *information snippets* that are extracted from documents of varying data sources as elemental units of information. Let V be the set of all available information snippets. We make the following assumptions for the structure of these snippets: First, every snippet $v_i \in V$ is associated with a timestamp t_i which records when the event(s) in the snippet occurred in the real-world. Second, it is associated with a data source s_i. Data sources may correspond to newspapers such as the New York Times or the Wall Street Journal but can encompass all digital media that provide event-based information. Last, every snippet has a content that describes the characteristics of the snippet. This content may vary depending on the application scenario and the available extraction possibilities. For STORYPIVOT in its current state, we use a black box extraction mechanism as visualized in Figure 1(a). Specifically, our extraction pipeline works as follows: It first collects textual excerpts from documents found on EventRegistry [9], i.e., it extracts the documents and breaks their text down based on paragraphs, title, etc. These excerpts are then forwarded to Open Calais which is an open-source annotation tool [16]. This tool provides additional information if available, for example on entities or keywords associated with the excerpt. The original text analyzed for the excerpt as well as its corresponding annotations thus form the content of the snippet.

Once snippets have been correctly extracted, the problem of finding stories is to search V for all snippets $v_j \in V$ that form a story c_ℓ. Let C be the set of all stories that can be extracted from the snippets in V. To process information snippets, we adopt a two-phase strategy which first determines the stories $C_i \subseteq C$ that can be found in a single source s_i. To compute these stories, V is partitioned into subsets $V_i \subseteq V$ where subscript i corresponds to the data source s_i. These sets of information snippets are then processed independently of each other to identify the stories within s_i. A visualization of this phase is shown in Figure 1(b) per data source. For example, for every snippet v_i^1 found in data source s_1 and thus part of V_1, story identification proceeds as follows: It first determines the most likely story that v_i^1 can be matched to. If no such story exists, it generates a new story around this information snippet, otherwise it adds the snippet to its best matching story. After all snippets in V_1 have been evaluated, C_1 consists of $c_1^1, \ldots, c_m^1$ source-specific stories. The evolution of stories, and thus implicitly the relationship of snippets, depends on the evolution of the corresponding real-world events which have to be captured correctly. In that context, we observe that it is possible for stories to split into multiple substories or to merge into a bigger story. For example political and economic events were interwoven during the height of the Ukraine crisis while they started to separate after the situation had (temporarily) stabilized. In order to capture the evolution of stories, we incrementally construct stories [5] instead of using single pass story detection [1, 17].

Once story identification for all available sources has been completed, we align stories across data sources. The alignment step is crucial for story detection for two reasons. First, it allows users of STORYPIVOT to retrieve unbiased information, i.e., a complete view of all available information on a story from a variety of sources instead of a single perspective which is the case if data is retrieved from a single source. Second, it allows us to correct errors made during the story identification phase. In Figure 1, we show that v_4^1 has been wrongly assigned to story c_1^1. Story alignment helps to determine this mistake by correlating events across sources and finding irregularities in the matching process which are then corrected. We call this correction step story refinement (Figure 1(d)). More detailed descriptions of the story identification and story alignment phases are presented next. Finally, we point out that the proposed two-level approach allows us to handle changes to the set of available data sources efficiently. As new sources become available, we first identify the stories associated with them and then align them with existing stories extracted from previously available sources. This enables an efficient integration of new data sources into the established information flow.

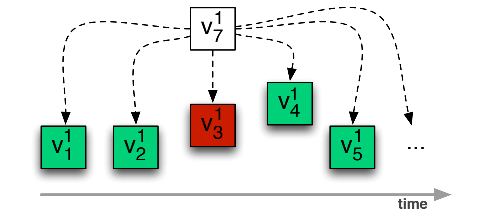
2.2 Story Identification

To enable efficient information aggregation into a story, we propose an incremental solution for the problem of story identification. As explained previously, story identification is the task of finding those information snippets $v_j^i \in V_i$ that refer to the same story c_ℓ. Recall that an information snippet is described by its content, its data source, and its timestamp. The evolution of a story and the connectivity of snippets can thus be described by matching the content of different snippets along its temporal axis. If a snippet v_j^i is sufficiently similar to any other candidate snippets $v_k^i \in V_i$ they may be part of the same story. Analogously, if the content of v_j^i and v_k^i is sufficiently dissimilar, it is intuitive to assume that they belong to different stories.

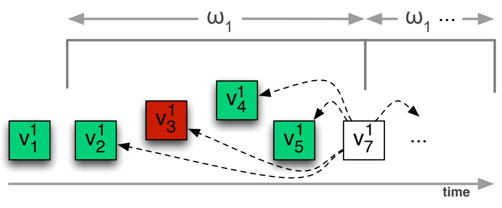
For this framework, we explore two different matching techniques for two snippets v_j^i and v_k^i: The first mechanism follows a *complete* matching scheme, while the second one pursues a *temporal* scheme as illustrated in Figure 2. The complete mechanism serves as a baseline for our system as we observe that due to the evolving nature of stories, complete mechanisms overfit stories as they tend to add related snippets to the same story independently of the evolution of the story in between. In reality, story evolution means that characteristics of a story change over time. Consider the example shown in Figure 2. Imagine that snippet v_7^1 is compared with the existing story set C_1: Here, it is not useful to compare it against all events $v_1^1 \ldots v_6^1$ associated with previously found stories (Figure 2a) because an event that happened a long time before v_7^1 can have completely different content, even if they both refer to the same entities. Take as an example the Ukraine crisis which started with civilian protests and evolved into a military conflict. Comparing snippets describing the fights around Donezk and protests on the Independence Square will not likely yield good results. In contrast, if we consider events closer to the fights such as the plane crash or the split of Crimea, there will be a higher overlap in entities and description between the single data snippets [12]. To leverage temporal information, we use a sliding window based technique for story identification. Instead of comparing v_7^1 to all other snippets, we thus reduce the window of comparison to all snippets $v_k^1 \in V_1$ for which the corresponding timestamp t_k^1 is in a range of $t_7^1 - \omega_1 \leq t_k^1 \leq t_7^1 + \omega_1$ (Figure 2b). Apart from providing a more realistic presentation of story evolution, sliding windows guarantee efficiency in information processing as the search space for related information snippets is reduced effectively.

2.3 Story Alignment

The story identification phase concludes with producing a set of stories C_i that contain all stories that have been identified for a data source s_i. These stories can then be used for cross-source story alignment, i.e., finding those stories in a data source s_i that contain the same semantic information as the stories in a different data source s_j. Imagine that there are two sources s_1, corresponding to the New York Times, and s_n, corresponding to the Wall Street Journal. Though both of these information sources provide details on events happening in the United States, they have different perspectives on topics expressed in their published documents. Here specifically, this difference is due to political orientation or different concepts for content presentation in the newspaper, amongst others. Story alignment helps to enrich the stories identified for each data



(a) Complete story identification.



(b) Temporal story identification.

Figure 2: Story identification execution modes.

source and enables providing end users with a more encompassing view of a story. This means that users are not only provided with source-specific and thus potentially biased information but have a structured way of exploring all perspectives of a story. These perspectives can be realized for stories that evolve over time but also for snippets capturing only one specific real-world event that can be correlated across data sources. In that sense, story alignment enhances the results found during the initial story identification phase.

For story alignment, we observe that two stories are likely to refer to the same real-world story if their evolution is similar and their content is similar as well. It is highly unlikely that two stories c_1 and c_2 are similar if c_1 ends at time t_i and c_2 starts at t_j with $t_i \ll t_j$. We observe that in the alignment phase, we can partition information snippets into two categories considering their purpose for alignment: they can (a) help to align stories or (b) enrich stories. The first type is used for the actual alignment step as two snippets v_k^i and v_l^j from sources s_i and s_j provide aligning information if they contain similar entities, description, and are temporally (approximately) similar. If data snippets enrich a story instead, they provide additional information that is contained in data source s_i but not in s_j. Relevant examples include special reports, background information etc. that are produced by journalists working for one data source but not the other.

Once stories have been compared across sources, the last step of story alignment is to resolve conflicts between the results of story identification and story alignment. Consider the example shown in Figure 1. Here, story alignment reveals a mismatch in the story results of data sources s_1 and s_n: While v_1^1, v_2^1, v_4^1, and v_5^1 belong to the same story in s_1 (c_1^1), their related snippets are in different stories in s_n (c_1^n and c_3^n). After story alignment, the system contains for this example two integrated stories, c_1' and c_3'. Resolving conflicts between results of these two phases helps to refine the results of the story identification phase as the decisions made during story alignment can be propagated back into the story sets of data sources if desired. Nevertheless, even if a story cannot be aligned across different data sources, it is still going to be present in the result set after story alignment has been executed. That is, if stories appear only within a data source, they may still hold interest for a variety of users. Imagine for example ten data sources out of which nine focus on business data and one covers sports events. Users then may search for a specific sports club which will require StoryPivot to return both sports and business stories independently of how many data sources actually report each story.

2.4 Dynamic Interaction between Story Identification and Alignment

So far, we talked about batch processing a set of information snippets abstractly. If these snippets are extracted from online newspapers, blogs, etc., they are generated dynamically every time a news document is published online. Furthermore, there exist different types of data sources which do not necessarily publish their information in a temporally ordered manner. Local events for example usually get picked up faster by local newspapers and might be delayed in international media because of locality and interest in those events. To address these challenges, story identification and alignment need to be dynamically integrated and realized efficiently as to provide users with live information on ongoing stories. This integration needs to consider an efficient representation of stories to reduce the search space and thus provide near real-time integration and out-of-order integration of events into evolving stories. To achieve efficient integration of story identification and alignment, we propose to abstract from snippets and stories into one common format which we refer to as a *sketch* [13]. A sketch is a (smaller) unified representation of the snippet or story that allows for fast and efficient similarity comparisons between stories and snippets or amongst stories across different sources. Last, notice that data sources are not always known to end users. Hence any story detection system should allow the addition or removal of data sources. The goal of our work is to address all of these problems with temporal story evolution techniques that adhere to the principles and ideas mentioned in the previous sections.

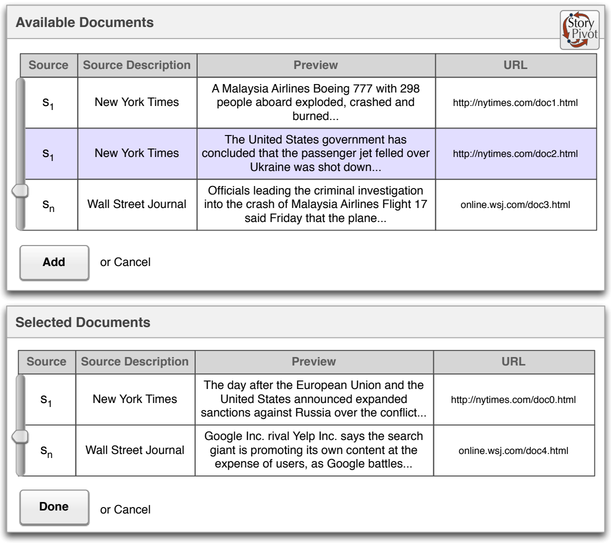
3. USE CASES

StoryPivot has practical utility for a variety of users. In this section, we present two target groups of users of our system and describe specific scenarios in which they can employ StoryPivot to obtain information about stories that they are interested in. For both of these user types, we envision an interface where the users can explore stories and where our system supports different patterns of accessing information.

Expert Scientist. Given its two-phase construction of stories, our system can provide users with insights on a story from either within a source or aggregated across sources. This functionality is essential for example for political scientists or journalists who want to determine the evolution of conflicts or interactions: Examining data within a data source reveals source bias and explicitly contrasts the different points of view that arise around a specific story. The results from the story alignment phase then help to construct a complete view of a story and to put single pieces of information from different data sources into context.

Casual Reader. As a casual news reader, it is often difficult to obtain a clear picture of the reason for stories to evolve the way they do. For example, young news readers are often not fully aware of the historical developments and trends that influence politics. Also, readers might get interested in a story only after it has gained some traction in the media. Investigating the timeline of a story can help such users to get a comprehensive view of which events contribute to a story and to understand the implications certain events have on the development of this story.

In addition to these inherent features of our system, we can further extend it with interfaces to existing knowledge bases such as DBpedia [3]. Connecting StoryPivot to knowledge bases explicitly helps experts and casual users to obtain more information on the context of stories and to explore the data provided by our framework more extensively.



Available Documents

Source	Source Description	Preview	URL
s_1	New York Times	A Malaysia Airlines Boeing 777 with 298 people aboard exploded, crashed and burned...	http://nytimes.com/doc1.html
s_1	New York Times	The United States government has concluded that the passenger jet felled over Ukraine was shot down...	http://nytimes.com/doc2.html
s_n	Wall Street Journal	Officials leading the criminal investigation into the crash of Malaysia Airlines Flight 17 said Friday that the plane...	online.wsj.com/doc3.html

Selected Documents

Source	Source Description	Preview	URL
s_1	New York Times	The day after the European Union and the United States announced expanded sanctions against Russia over the conflict...	http://nytimes.com/doc0.html
s_n	Wall Street Journal	Google Inc. rival Yelp Inc. says the search giant is promoting its own content at the expense of users, as Google battles...	online.wsj.com/doc4.html

Figure 3: Document selection module.

4. DEMONSTRATION

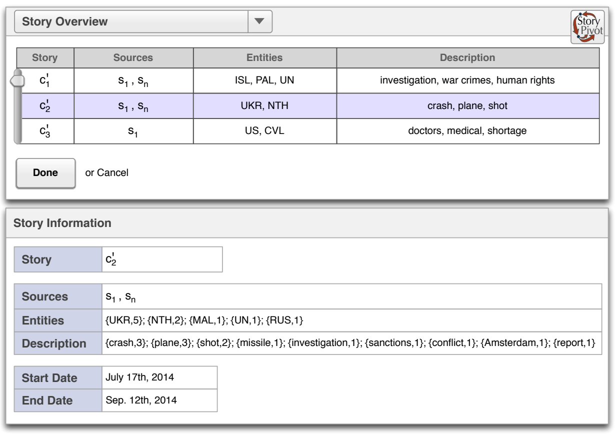
In this section, we describe how we structure our demonstration and present the functionalities that will be available when interacting with the StoryPivot system.

4.1 Implementation

Our demonstration introduces StoryPivot, a system that efficiently processes event datasets to find semantically correlated events that form stories, i.e., that describe relationships between entities with evolving content. Users of StoryPivot can explore two different views on stories following the differentiation between story identification within a data source and alignment across data sources presented in Section 2. The backend processing of information extracted from documents is done as shown in Figure 1: First, we use an extraction mechanism that retrieves snippets of information which is followed by our implementation of the story identification and alignment steps. To realize story identification, we implemented the two different identification models shown in Figure 2, temporal story identification and complete story identification. For story alignment, we extend the techniques used in story identification to consider matching stories with each other instead of comparing a snippet to a story allowing for more tolerance in the temporal alignment of stories. The implemented methods can be combined on the fly by the user to test advantages and disadvantages as well as compare result quality for these varying techniques. The event data explored for this demonstration is taken from real-world online newspapers and blogs such as the New York Times and the Wall Street Journal as captured by existing event repositories such as GDELT [11] or EventRegistry [9].

4.2 Functionalities

For the demonstration, we will provide dynamic and static execution components. First, we will show how stories evolve on a small scale with a predefined example that the user can explore by adding and removing documents from the system. This will enable



Story	Sources	Entities	Description
c'_1	s_1 , s_n	ISL, PAL, UN	investigation, war crimes, human rights
c'_2	s_1 , s_n	UKR, NTH	crash, plane, shot
c'_3	s_1	US, CVL	doctors, medical, shortage

Story Information

Story	c'_2
Sources	s_1 , s_n
Entities	{UKR,5}; {NTH,2}; {MAL,1}; {UN,1}; {RUS,1}
Description	{crash,3}; {plane,3}; {shot,2}; {missile,1}; {investigation,1}; {sanctions,1}; {conflict,1}; {Amsterdam,1}; {report,1}
Start Date	July 17th, 2014
End Date	Sep. 12th, 2014

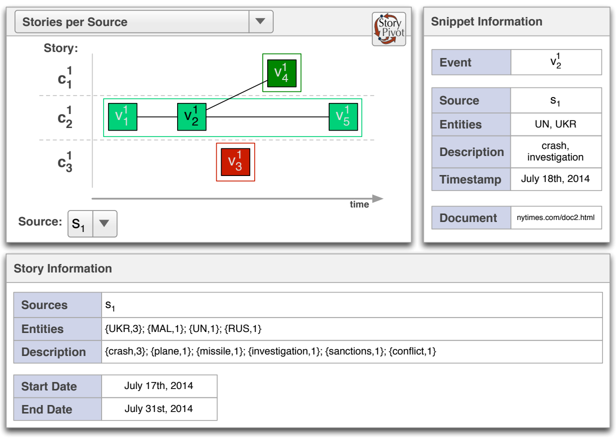
Figure 4: Story overview module.

users to understand how extracted information from a document changes the outcome of the story identification and alignment phases. Furthermore, this part of our demonstration serves to illustrate the applicability and output quality of our system. To understand the actual performance of StoryPivot and to be able to compare it against existing approaches, we will provide users with manually curated stories taken from well-known news providers. Furthermore, we also present examples of large scale story detection and examine the scalability and efficiency of StoryPivot when processing larger amounts of data. The purpose for this part of the demonstration is to show that real-time event integration can be achieved through efficient story identification and alignment mechanisms. Users will be able to explore the results of the larger integration run and can query StoryPivot to see the evolution of a story over time within and across sources. For simplicity, queries will consist of enquiries about specified real-world events or entities.

4.2.1 *Interactive Story Detection*

For this part of our demonstration, we will provide the user with an interface where a set of documents can be selected. Each of these documents is taken from real-world event data streams. An example for the document selection mechanism is shown in Figure 3. The data sources here are two well-known newspapers and documents correspond to news articles published on their corresponding websites. For this example specifically, the articles mainly focus on the downing of a plane in the Ukraine in July 2014. Once the user has selected a subset of these articles, an initial set of stories within the selected data sources is computed and then stories are aligned across data sources, as in Figure 4. After the computation has finished, the user will be able to explore the data in two different ways corresponding to the view (i) within and (ii) across data sources.

Stories per Source Module. The "Stories per Source" module, Figure 5, shows the computed stories within a source. Information snippets are sorted according to their story affiliation and can be browsed to understand why they are associated with the same or different stories. This module represents the story identification part of our work. Next to visualizing and explaining the correlation of snippets within a story, we also show the relationship of snippets across stories but within a data source. Take v_2^1 in Figure 5 as an example: We show that it is connected not only to v_1^1 and v_5^1 but also to v_4^1 which contains information on a call for investigation on some problems in the Israeli conflict around the same time as the jet's downing. As the entities for both snippets and the contents are similar, there exists such a connection intuitively though the events



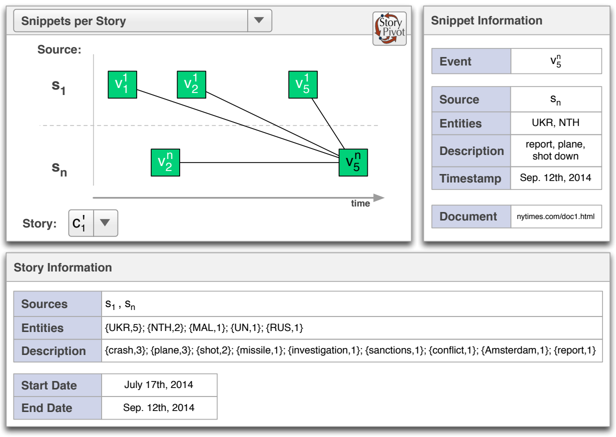
Snippet Information

Event	v_2^1
Source	s_1
Entities	UN, UKR
Description	crash, investigation
Timestamp	July 18th, 2014
Document	nytimes.com/doc2.html

Story Information

Sources	s_1
Entities	{UKR,3}; {MAL,1}; {UN,1}; {RUS,1}
Description	{crash,3}; {plane,1}; {missile,1}; {investigation,1}; {sanctions,1}; {conflict,1}
Start Date	July 17th, 2014
End Date	July 31st, 2014

Figure 5: Stories per source module. Displays (subset of) stories for a selected source.



Snippet Information

Event	v_5^n
Source	s_n
Entities	UKR, NTH
Description	report, plane, shot down
Timestamp	Sep. 12th, 2014
Document	nytimes.com/doc1.html

Story Information

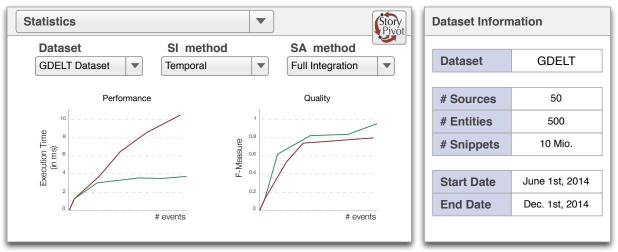
Sources	s_1 , s_n
Entities	{UKR,5}; {NTH,2}; {MAL,1}; {UN,1}; {RUS,1}
Description	{crash,3}; {plane,3}; {shot,2}; {missile,1}; {investigation,1}; {sanctions,1}; {conflict,1}; {Amsterdam,1}; {report,1}
Start Date	July 17th, 2014
End Date	Sep. 12th, 2014

Figure 6: Snippets per story module. Displays (subset of) information snippets for a selected story.

can be separated through their context. Here, v_2^1 is similar to v_1^1 and v_5^1 which indicates that it is in fact part of a different story than v_4^1. In addition to providing this overview information per snippet, we allow users of StoryPivot to explore snippets within a data source in this module. Changing the selected snippet will automatically change the displayed content.

Snippets per Story Module. As the second exploration module, we provide a "Snippets per Story" module, as in Figure 6, which shows a user-selected story that has been aligned across data sources. Again, this module will help the user to understand why our algorithms make certain decisions through visualizing the story alignment phase. System users can select a story that they are interested in and again explore single information snippets within the story to determine their connectivity. In the example of Figure 6, the selected snippet, v_5^n, is well-connected within story c'_1 which consists of information on the downing of the jet ranging from the initial news to results of the investigation.

Both Figure 5 and Figure 6 show snippet-centric views of story detection. We choose this form of visualization because it puts emphasis on the evolving nature of stories as a set of semantically connected pieces of information. To enable users to explore this evolution, additional documents can be added to the system after



Dataset: GDELT Dataset
SI method: Temporal
SA method: Full Integration

Dataset Information

Dataset	GDELT
# Sources	50
# Entities	500
# Snippets	10 Mio.
Start Date	June 1st, 2014
End Date	Dec. 1st, 2014

Figure 7: Statistics module. Displays information on experiments with larger datasets.

the initial import step. Furthermore, we allow users to remove documents from STORYPIVOT to explore how missing information affects the displayed stories.

4.2.2 *Large Scale Story Detection*

In this part of our demonstration, we want to show the effectiveness of our mechanisms for detecting story evolution on larger datasets. In contrast to the previously presented interactive components, we will not allow the user to add or remove single documents but present them with results from large-scale experiments run beforehand. Users are shown information and statistics on a set of extensive experiments through an overview module as visualized in Figure 7. Here, they can select all available datasets as well as different execution mechanisms that were run for these datasets. This will provide an intuition of which techniques are more advantageous than others in terms of data analysis but also performance over time.

Next to these static ways of exploring these datasets and the meta information on performance and quality, we also allow users to use the modules presented previously for an interactive exploration of these datasets. As a result, users develop an understanding of how stories evolve over time and how our system captures this evolution. It will not be possible to add or remove documents for this setup as the datasets will be fixed beforehand.

4.3 Take-away Points

With our work, we want to emphasize that story detection is an important and interesting problem that helps users to obtain information on real-world stories in a structured way. We use a two-step mechanism of story identification and story alignment for two reasons: First, it inherently guarantees that users can choose the granularity of stories presented to them. For example, they can decide to see a story from within a data source or as a (partial) integration across sources through our story alignment methods. With our interactive design, we want to show system users how these two mechanisms interact and help to build a better understanding of story evolution. Second, this two-step mechanism provides efficient implementation mechanisms and fast information processing because it makes effective use of incremental rather than complete computation of results. Remember that stories evolve over time which means that using and integrating all information snippets completely is more harmful than information processing in a time-aware manner. The stability and performance of our mechanisms are shown in large-scale experiments which help to obtain a better understanding of the efficiency of STORYPIVOT.

5. CONCLUSION

In this demonstration proposal, we have introduced a system called STORYPIVOT that constructs stories from information snippets that were extracted from news documents such as online blog entries, articles, reports, and others. We proposed to process this data in two steps through story identification and subsequently story alignment. Both of these can be interactively explored in our demonstration with a small predefined example which allows users to manipulate stories by adding and removing information at will. In the second part of our demonstration, we examine large scale datasets to show that the mechanisms devised for this system are stable over time in terms of performance and result quality. Furthermore, we allow users to explore the results of these large-scale experiments interactively to understand how STORYPIVOT forms stories.

6. REFERENCES

[1] J. Allan, R. Papka, and V. Lavrenko. On-line New Event Detection and Tracking. *SIGIR*, pages 37–45, 1998.
[2] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava. Dense Subgraph Maintenance Under Streaming Edge Weight Updates for Real-time Story Identification. *PVLDB*, 5(6):574–585, 2012.
[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A Nucleus for a Web of Open Data. *ISWC*, pages 722–735, 2007.
[4] X. L. Dong, B. Saha, and D. Srivastava. Less is More: Selecting Sources Wisely for Integration. *PVLDB*, pages 37–48, 2013.
[5] A. Gruenheid, X. L. Dong, and D. Srivastava. Incremental Record Linkage. *PVLDB*, 7(9):697–708, 2014.
[6] J. I. Haidar. Sanctions and Trade Diversion: Exporter-Level Evidence from Iran. *ERF Annual Conference: Economic Development and Social Justice*, 2014.
[7] J. Hinz. The Ties that Bind: Geopolitical Motivations for Economic Integration. *Sixteenth Annual Conference of The European Trade Study Group ETSG*, 2014.
[8] Y. Keneshloo, J. Cadena, G. Korkmaz, and N. Ramakrishnan. Detecting and Forecasting Domestic Political Crises: A Graph-based Approach. *WebSci*, pages 192–196, 2014.
[9] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event Registry: Learning About World Events from News. *WWW Companion*, pages 107–110, 2014.
[10] P. Lee, L. V. S. Lakshmanan, and E. E. Milios. Incremental Cluster Evolution Tracking from Highly Dynamic Network Data. *ICDE*, pages 3–14, 2014.
[11] K. Leetaru and P. Schrodt. GDELT: Global Data on Events, Language, and Tone, 1979-2012. *Inter. Studies Association Annual Conf.*, 2013.
[12] P. Li, X. L. Dong, A. Maurino, and D. Srivastava. Linking Temporal Records. *PVLDB*, 4(11):956–967, 2011.
[13] S. Muthukrishnan. *Data Streams: Algorithms and Applications*. Now Publishers Inc, 2005.
[14] N. Ramakrishnan et al. 'Beating the News' with EMBERS: Forecasting Civil Unrest Using Open Source Indicators. *SIGKDD*, pages 1799–1808, 2014.
[15] T. Rekatsinas, X. L. Dong, L. Getoor, and D. Srivastava. Finding Quality in Quantity: The Challenge of Discovering Valuable Sources for Integration. *CIDR*, 2015.
[16] T. Reuters. Calais. *http://www.opencalais.com/*, 2008 – 2014.
[17] H. Sayyadi, M. Hurst, and A. Maykov. Event Detection and Tracking in Social Streams. *ICWSM*, 2009.
[18] S. Schutte and K. Donnay. Matched Wake Analysis: Finding Causal Relationships in Spatiotemporal Event Data. *Political Geography*, 41:1–10, 2014.
[19] P. Upadhyaya, M. Unutzer, M. Balazinska, D. Suciu, and H. Hacigumus. Affordable Analytics on Expensive Data. *Data4U*, page 19, 2014.
[20] M. D. Ward, N. W. Metternich, C. L. Dorff, M. Gallop, F. M. Hollenbach, A. Schultz, and S. Weschle. Learning from the Past and Stepping into the Future: Toward a New Generation of Conflict Prediction. *International Studies Review*, 15(4):473–490, 2013.
[21] N. Weller and K. McCubbins. Raining on the Parade: Some Cautions Regarding the GDELT Dataset. *Blog Post on Political Violance a Glance*, 2014.